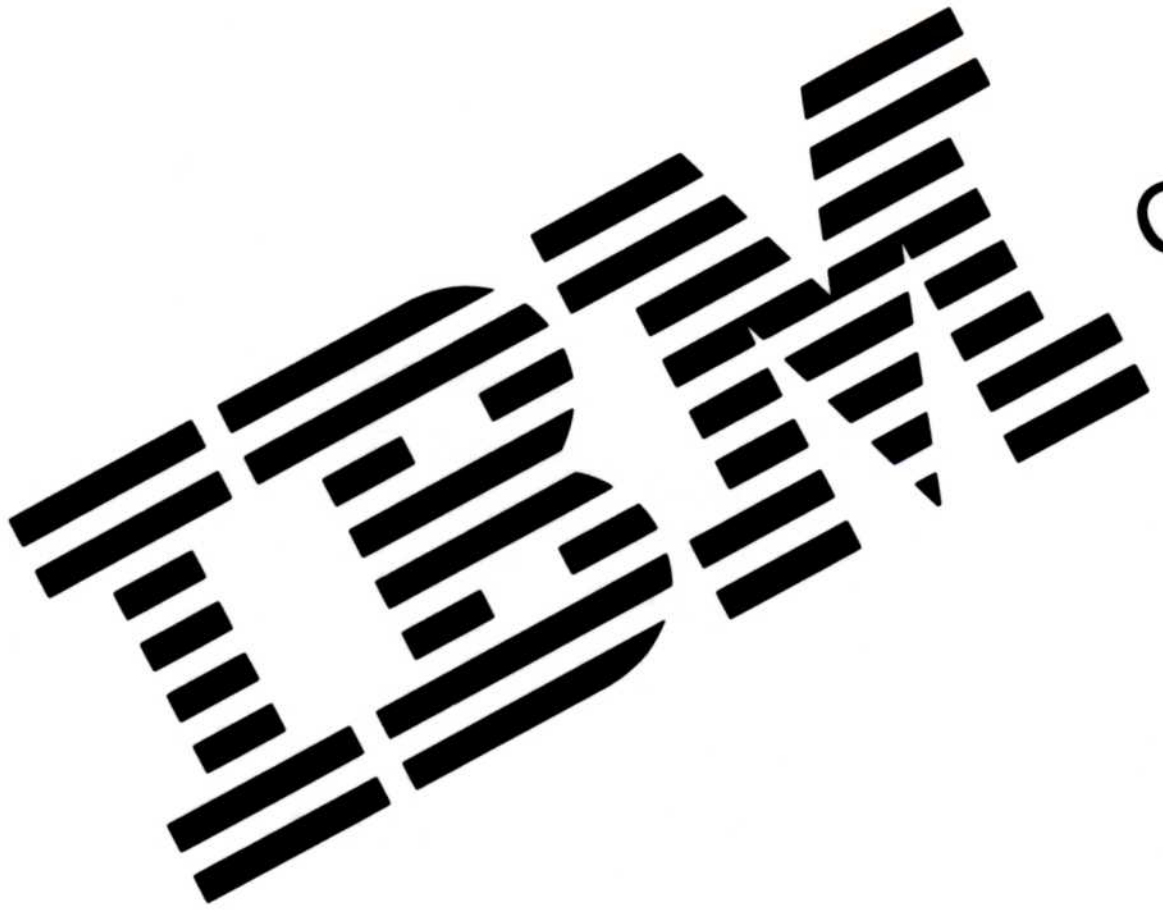


Operating System/2™ Standard Edition

User's Reference

Programming Family

First Edition (September 1987)

The following paragraph does not apply to the United Kingdom or any country where such provisions are inconsistent with local law:
INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This publication could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time.

It is possible that this publication may contain reference to, or information about, IBM products (machines and programs), programming, or services that are not announced in your country. Such references or information must not be construed to mean that IBM intends to announce such IBM products, programming, or services in your country.

Requests for copies of this publication and for technical information about IBM products should be made to your IBM Authorized Dealer or your IBM Marketing Representative.

Operating System/2 and OS/2 are trademarks of the International Business Machines Corporation.

© Copyright International Business Machines Corporation 1987

All rights reserved. No part of this publication may be reproduced or distributed in any form or by any means without prior permission in writing from the International Business Machines Corporation.

Preface

This book contains user information for IBM Operating System/2™ (OS/2™). It provides information on the use of OS/2 commands and shows how to use them in two separate environments: the OS/2 multitasking environment and an environment that is similar to the IBM Personal Computer Disk Operating System Version 3.3. The commands are presented in three alphabetic groups:

- OS/2 internal and external commands
- Commands for batch files
- Statements for the CONFIG.SYS file.

Operating System/2 Diskettes

There are four Operating System/2 diskettes:

- *Operating System/2 Diskette 1* contains the files necessary to start the operating system and the Program Selector. The system can be started by inserting this diskette in Drive A and then pressing Ctrl+Alt+Del.
- *Operating System/2 Diskette 2* and *Operating System/2 Diskette 3* contain OS/2 external commands, device drivers, and related support files.
- *Operating System/2 Installation Diskette* contains the installation program that installs OS/2. The installation program is started by inserting this diskette in drive A and then pressing Ctrl+Alt+Del.

IBM Operating System/2 and OS/2 are trademarks of the International Business Machines Corporation.

Related Publications

This book is intended for the application program user. It is assumed that you have read the *IBM Operating System/2 User's Guide*, and that OS/2 is installed on your computer.

Other books related to OS/2 are:

IBM Operating System/2:

- *Programmer's Guide*
- *Technical Reference*.

Contents

Chapter 1. Introduction 1-1
Two Modes of Operation 1-1
Commands for One Mode 1-1
Running OS/2 Programs 1-2
Storage Allocation in OS/2 Mode 1-3
Running DOS Programs 1-3
What Happens When You Start the System 1-4

Chapter 2. OS/2 Commands 2-1
Internal and External Commands 2-1
Reading the Command Diagram 2-2
Commands for Problem Determination 2-6
ANSI (extended keyboard and display) 2-7
APPEND (search path) 2-8
ASSIGN (assign drives) 2-12
ATTRIB (file attribute) 2-14
BACKUP (back up files) 2-17
BREAK (set) 2-20
CHCP (change code page) 2-22
CHDIR (change current directory) 2-26
CHKDSK (check disk) 2-28
CLS (clear screen) 2-31
CMD (command processor) 2-32
COMMAND (command processor) 2-34
COMP (compare files) 2-36
COPY (copy files) 2-39
CREATEDD (create dump diskette) 2-43
DATE (set system date) 2-45
DETACH (process noninteractively) 2-47
DIR (display files in directory) 2-49
DISKCOMP (compare diskettes) 2-53
DISKCOPY (copy diskettes) 2-56
DPATH (search path) 2-59
ERASE (delete files) 2-61
EXIT (end a command processor) 2-63
FDISK (manage partitions) 2-64
FIND (filter) 2-67
FORMAT (a fixed disk or diskette) 2-70
GRAFTABL (graphics) 2-74

HELP (help with messages) 2-76
JOIN (join drives) 2-79
KEYB (keyboard layout) 2-81
LABEL (volume label) 2-84
MKDIR (make directory) 2-86
MODE (device control) 2-88
MORE (filter) 2-97
PATCH 2-98
PATH (search path) 2-101
PRINT 2-103
PROMPT (command prompt) 2-105
RECOVER (recover files) 2-107
RENAME (rename a file) 2-109
REPLACE (replace files) 2-110
RESTORE (restore backed up files) 2-112
RMDIR (remove directory) 2-115
SET (environment variables) 2-117
SETCOM40 (set COM port address) 2-121
SORT (filter) 2-124
SPOOL (print queue) 2-126
START (programs automatically) 2-130
SUBST (substitute drives) 2-133
SYS (transfer hidden files) 2-135
TIME (set system time) 2-137
TRACE (system events) 2-139
TRACEFMT (trace formatter) 2-141
TREE (display directory structure) 2-143
TYPE (display file contents) 2-144
VER (display OS/2 version number) 2-146
VERIFY (set data verification) 2-147
VOL (display disk volume label) 2-149
XCOPY (copy subdirectories) 2-151

Chapter 3. Commands for Batch Files 3-1
Using Batch Files 3-1
Creating Batch Files 3-2
Step-By-Step Instructions 3-2
Running a Batch File 3-2
Batch Processing Differences 3-3
Special Batch Files 3-4
OS2INIT.CMD File 3-4
AUTOEXEC.BAT File 3-5
STARTUP.CMD File 3-5

Using Replaceable Parameters in a Batch File 3-8
Defining Replaceable Parameters with Names 3-9
Suppressing Echoing of Batch File Statements 3-10
Batch Commands 3-10
CALL (nest batch file) 3-11
ECHO 3-13
ENDLOCAL (environment) 3-15
EXTPROC (external processor) 3-17
FOR 3-18
GOTO 3-20
IF 3-21
PAUSE 3-24
REM (remarks) 3-26
SETLOCAL (environment) 3-27
SHIFT 3-28

Chapter 4. Statements for the CONFIG.SYS File 4-1
The CONFIG.SYS File 4-1
The Installation Program 4-2
Changing CONFIG.SYS 4-2
Code Page Switching 4-3
Statements for CONFIG.SYS 4-4
BREAK 4-5
BUFFERS 4-7
CODEPAGE (prepare code pages) 4-8
COUNTRY 4-10
DEVICE 4-13
ANSI.SYS Device Driver 4-16
COM0x.SYS Device Driver 4-17
EGA.SYS Device Driver 4-19
EXTDSKDD.SYS Device Driver 4-20
MOUSExxx.SYS Device Driver 4-24
POINTDD.SYS Device Driver 4-27
VDISK.SYS Device Driver 4-28
DEVINFO (code page device) 4-30
DISKCACHE 4-34
FCBS (file control blocks) 4-36
IOPL (input/output privilege) 4-38
LIBPATH (library search path) 4-39
MAXWAIT (maximum wait) 4-41
MEMMAN (storage allocation) 4-42
PAUSEONERROR (error prompting) 4-44
PRIORITY 4-45

PROTECTONLY (mode selection) 4-46
PROTSHELL (user interface) 4-47
REM (remarks) 4-49
RMSIZE (DOS mode size) 4-50
RUN 4-51
SHELL (DOS command processor) 4-53
SWAPPATH 4-55
THREADS 4-56
TIMESLICE 4-57
TRACE 4-59
TRACEBUF (trace buffer) 4-61

Chapter 5. Making the Command Processor Work for You 5-1
Redirecting Input and Output 5-1
Piping Output to Another Command 5-2
Processing Commands Conditionally 5-3
Separating and Grouping Commands 5-4
Redirection in OS/2 Mode 5-5
Redirection Sequences 5-6
Examples of Redirected I/O 5-7
Running Noninteractive Programs 5-9
Preventing All Output by a Program 5-9
Echoing of Redirection Statements in OS/2 Mode 5-10
Allowing Symbols to be Input as Text 5-10

Appendix A. A Quick Reference to OS/2 Commands A-1
Commands for One Mode Only A-1

Appendix B. DOS Compatibility B-1
Compatibility Exceptions for Commands B-1
Guidelines for Running DOS Programs B-3
Running Batch Files B-3
Installing and Running Copy-Protected DOS Application Programs B-3
Using the SPOOL Command B-4
Using Serial Printers and Plotters B-4
Setting Asynchronous Communications Modes B-5
Redirecting Parallel Printer Output to a Serial Device B-5
Using Serial Printers and Plotters with DEVICE=COM0x.SYS B-6
Managing Applications Using Serial Devices in DOS Mode . B-6
Using a Mouse B-7
Using Applications With Enhanced Graphics Features B-8
System Limitations B-8

Appendix C. The Line Editor (EDLIN) C-1
How to Start the EDLIN Program C-1
Editing an Existing File C-1
Editing a New File C-2
Editing a Binary File C-3
The EDLIN Command Parameters C-4
The EDLIN Commands C-6
A (append lines) C-8
C (copy lines) C-9
D (delete lines) C-10
Edit Line C-13
E (end edit) C-15
I (insert lines) C-16
L (list lines) C-19
M (move lines) C-22
P (page) C-23
Q (quit edit) C-24
R (replace text) C-25
S (search text) C-28
T (transfer lines) C-31
W (write lines) C-32

Appendix D. Keyboard Templates D-1
Multiple Character Keys D-1

Appendix E. Accented Characters E-1
Nonescaping Key Combinations E-1

Appendix F. System Code Pages F-1
Using Code Pages F-1
Code Page 437 (U.S. IBM PC) F-2
Code Page 850 (Multilingual) F-3
Code Page 860 (Portuguese) F-4
Code Page 863 (Canadian-French) F-5
Code Page 865 (Nordic) F-6

Appendix G. Messages G-1
Device Error Message Format G-1
Device Error Responses G-2
Messages G-3

Appendix H. OS/2 Diskettes and Directories H-1
Operating System/2 Diskettes H-1

Directories Created by the Installation Program H-1

Glossary X-1

Index X-5

Chapter 1. Introduction

This chapter describes the two operating modes provided by IBM Operating System/2™ (OS/2™) and explains how their differences affect the processing of programs and the commands you can use.

Two Modes of Operation

IBM Operating System/2 has two operating modes:

- OS/2
- DOS

OS/2 mode refers to the multitasking environment that allows you to run more than one program written for OS/2 at a time. DOS mode refers to the environment that is similar to IBM Disk Operating System, Version 3.3.

Commands for One Mode

Each mode has its own command processor to interpret and run operating system commands. Most of the commands work in both modes. However, there are some that provide functions specific to one mode. In this book, symbols identify these commands:

OS/2 only 

DOS only

If a command does not have a symbol, this means that the command works in both modes. These symbols are also used to identify information pertaining to only one mode. If you enter a command in the wrong mode, the command processor does not recognize the command, and an error message is displayed. For example, suppose you enter the APPEND command, which has a DOS symbol, while at an OS/2 command prompt. You receive this message:

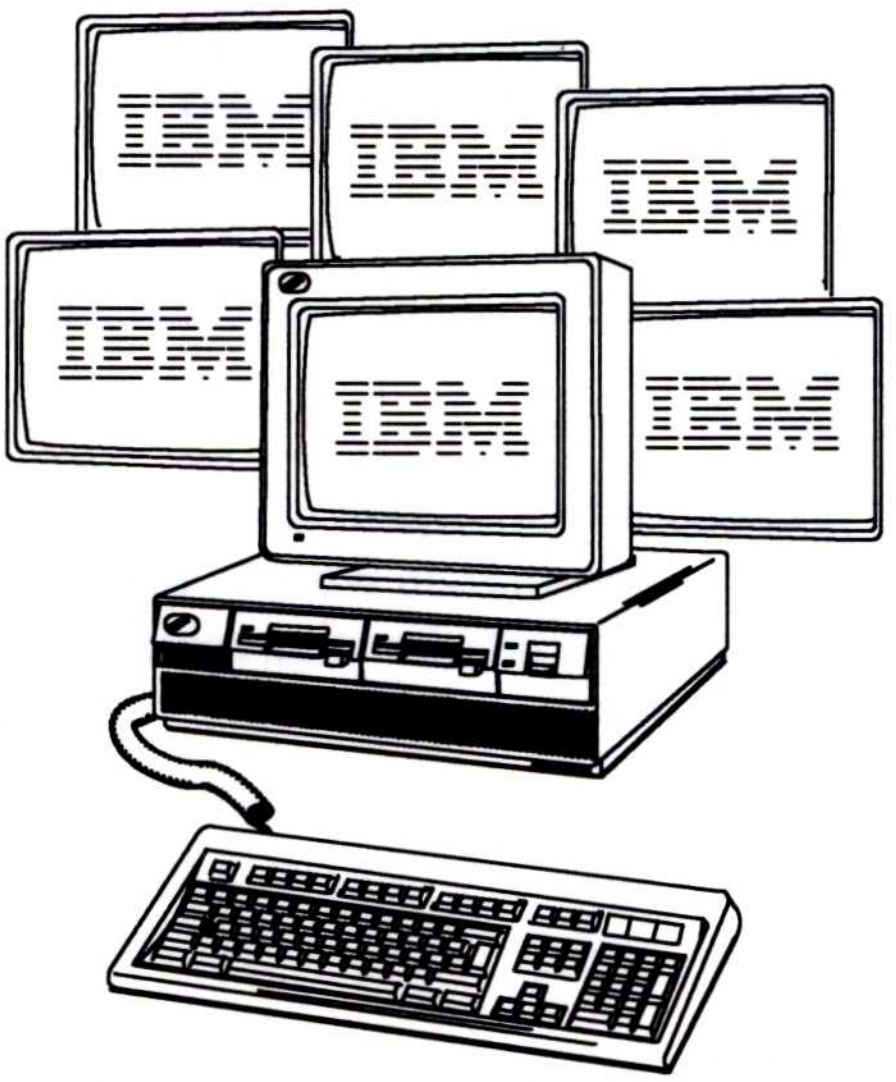
```
SYS0193:  \OS2\APPEND.COM cannot be run in
the IBM Operating System/2 mode.
```

If you are interested in an overview of the commands and their functions, refer to Appendix A, "A Quick Reference to OS/2 Commands" on page A-1. The symbols show you the commands that are unique to OS/2 or DOS mode.

There are some IBM Disk Operating System Version 3.3 commands that are not supported or are slightly different in DOS mode. See "Compatibility Exceptions for Commands" on page B-1 for these differences.

Running OS/2 Programs

In OS/2 mode, multiple sessions can be started, each with its own OS/2 command processor, to run programs or process commands.



Each OS/2 session runs independently of any other OS/2 session. If you are at the command prompt and change environment variables, they are known only to the current OS/2 session. For example, suppose you run a batch file that changes the paths set with PATH and DPATH commands, or the command prompt set with the PROMPT command. These changes are in effect only for the current session and any programs started in it, including additional OS/2 command processors started by the CMD command.

Storage Allocation in OS/2 Mode

OS/2 programs can use up to 16Mb of system storage installed in the computer. This means you can have a system or even a single program use far more than the 640Kb of system storage to which IBM Personal Computer DOS systems have traditionally been limited.

A storage allocation feature called *storage overcommitment* lets OS/2 programs use even more than the amount of system storage in the computer. This capability delivers storage on demand to any program or system configuration, using the following methods.

Discarding segments: When a program stops using a certain segment of storage, OS/2 discards the segment. The storage that has been freed up can then be used for other work.

Swapping segments: Some segments cannot be discarded when not needed, since the program may have changed some data in them. These segments are written to the disk when the space for them is needed by other segments. This is referred to as segment *swapping*. When the program needs the segment that was written to the disk, OS/2 loads it back into system storage.

Combining segments: Discarding and swapping segments, as just described, fragments storage and creates blank areas where segments have been discarded. If a program or system task needs a large piece of storage, OS/2 can move blocks of storage around so that unused areas of storage can be combined into larger areas until OS/2 gets the amount it needs. This is referred to as *storage compaction*.

Running DOS Programs

DOS programs can run only within the first megabyte of system storage and are limited to using not more than 640Kb.

In the DOS mode, only one DOS command processor can process a command or program at a time. Thus, only one DOS program can run at a time, and it is processed only when it is displayed on your screen. For example, if you are working on a DOS program and you return to the Program Selector, the DOS program stops running. It

begins running again when you return to it. If the DOS program was keeping track of the time of day by counting clock ticks, it may have an incorrect time when it resumes.

Note: Not all programs written for IBM Personal Computer DOS will run in DOS mode. For information that may help you determine whether your programs will run successfully, see "Guidelines for Running DOS Programs" on page B-3.

What Happens When You Start the System

When you start OS/2, the system looks for a file named CONFIG.SYS in the root directory of the startup drive and interprets the statements within the file. If this file contains the statement PROTECTONLY=NO, your system is configured for both DOS and OS/2 modes. OS/2 and DOS device drivers specified with DEVICE statements are loaded. OS/2 device drivers can process requests from programs in both modes, however DOS device drivers can only process requests from programs in DOS mode.

If the system cannot find a CONFIG.SYS file, it assigns configuration default values. For information on configuring your system, see Chapter 4, "Statements for the CONFIG.SYS File" on page 4-1.

After the system processes the CONFIG.SYS file, it looks for a file called STARTUP.CMD and interprets the commands in the file. When you select the OS/2 command prompt from the Start a Program list on the Program Selector menu, the system looks for a file called OS2INIT.CMD. This file is created at system installation time. The OS/2 command processor interprets the commands in the file. If your system is configured for both modes, the first time you select the DOS command prompt from the Program Selector, the system looks for a file called AUTOEXEC.BAT. The DOS command processor interprets the commands in the file.

These special batch files are useful because they can contain commands and programs that are run automatically. For an explanation of these files, see "Special Batch Files" on page 3-4.

Chapter 2. OS/2 Commands

The IBM Operating System/2™ (OS/2™) commands are presented in alphabetic order starting on page 2-7 of this chapter. For an overview of the commands and their functions, see "OS/2 Commands" on page A-2.

Internal and External Commands

OS/2 has two types of commands:

- Internal commands
- External commands.

Internal commands run immediately because they are loaded with OS/2.

External commands are on the disk as program files. When an external command is entered, OS/2 searches for it in the current directory of the default or specified drive. If not found, OS/2 continues searching for it in the directories listed in the most recent PATH command.

Note: System installation creates two special batch files for you, each of which contains the necessary paths to OS/2 external commands and data files. When you use the Program Selector to start an OS/2 command processor, the paths are automatically set by the OS2INIT.CMD batch file. The first time you select the DOS command prompt to start a DOS command processor, the paths are automatically set by the AUTOEXEC.BAT batch file. See "Special Batch Files" on page 3-4 for more information.

Global filename characters and reserved device names are not allowed in a command name. You can use global filename characters only when specifying filenames and filename extensions as command parameters. There are some commands where global filename characters are not permitted in filename parameters. For example, the FIND command does not allow them. In these cases, this is noted in the command explanation. For more information on global filename characters and reserved names, see the *OS/2 User's Guide*.

Note: System installation creates directories called: C:\OS2, C:\OS2\INTRO, C:\OS2\INSTALL, and C:\SPOOL. If you want the benefits of system installation, do not use these names for files or directories. For more information on the directory structure created by system installation, see Appendix H, "OS/2 Diskettes and Directories" on page H-1.

You can type in as many as 128 characters (bytes) at either command prompt before pressing Enter.

For commands displaying a large amount of output, you can press the Ctrl+Num Lock keys to pause the display of the output. Press any character key to continue the display.

Note: If you are using an IBM Enhanced Keyboard, use the Pause key to pause the display of output.

Reading the Command Diagram

The command diagram shows you how to enter a command so it can be interpreted by the command processor. Items in a command diagram are called the command syntax.

The command name and items required to make the command work appear on the baseline. Items below the baseline are optional.

A baseline ending with an arrowhead means the command syntax is continued on the following line. A baseline starting with an arrowhead means the command syntax is continued from the line above. A vertical bar at the end of a baseline marks the end of the command syntax.

Variable parameters have different values, depending on what you enter. Variables are shown in lowercase italicized letters.

A keyword parameter has a predefined value. A command name is a keyword. Keywords are shown in capitalized letters. Note that a parameter can be a combination of a keyword and a variable.

Include all punctuation shown in the diagram, such as colons, commas, parentheses, and question marks. In command syntax, the

b symbol is used to indicate a space. In command examples, spaces are used for readability.

Commands and their parameters can be entered in uppercase or lowercase letters, or a combination of both. In this book, uppercase letters are used in command examples.

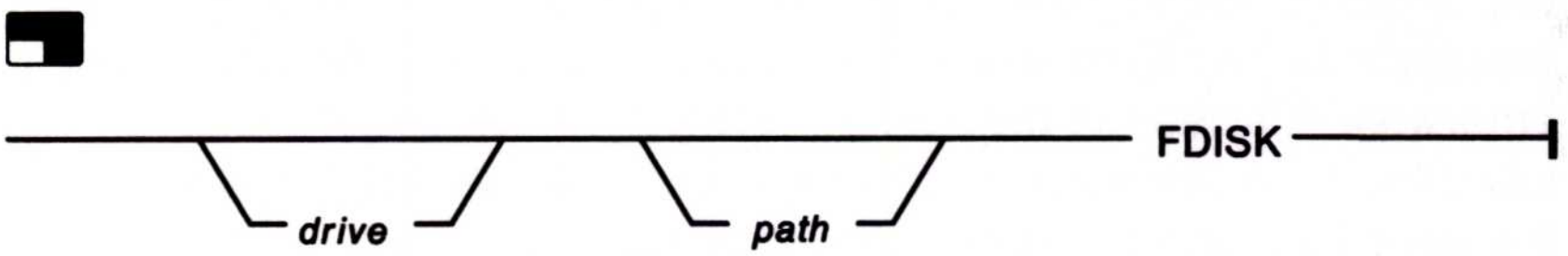
Internal and External Commands: If a command is an internal command, the command name is the first item in the syntax, as in this example:



Since EXIT has no parameters, to use it, just enter:

```
EXIT
```

If the command is an external program file, its name is preceded by an optional drive and path, as in this example:

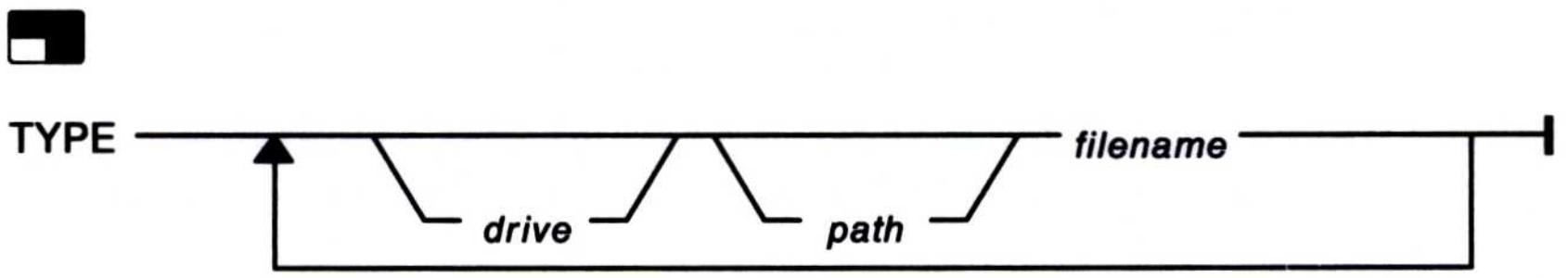


This means that you may have to enter the path to the directory that contains OS/2 external commands. For example, if the PATH command created by system installation for your OS2INIT.CMD batch file has been changed, and a path to OS/2 external commands no longer exists, you may have to enter:

```
C:\OS2\FDISK
```

Note that the symbol tells you this command can be used only at the OS/2 command prompt.

Specifying a Sequence More Than Once: An arrow that returns to a previous point on the baseline means that the sequence of items (such as drive, path, and filename) included by the arrow can be specified more than once, as in this example:



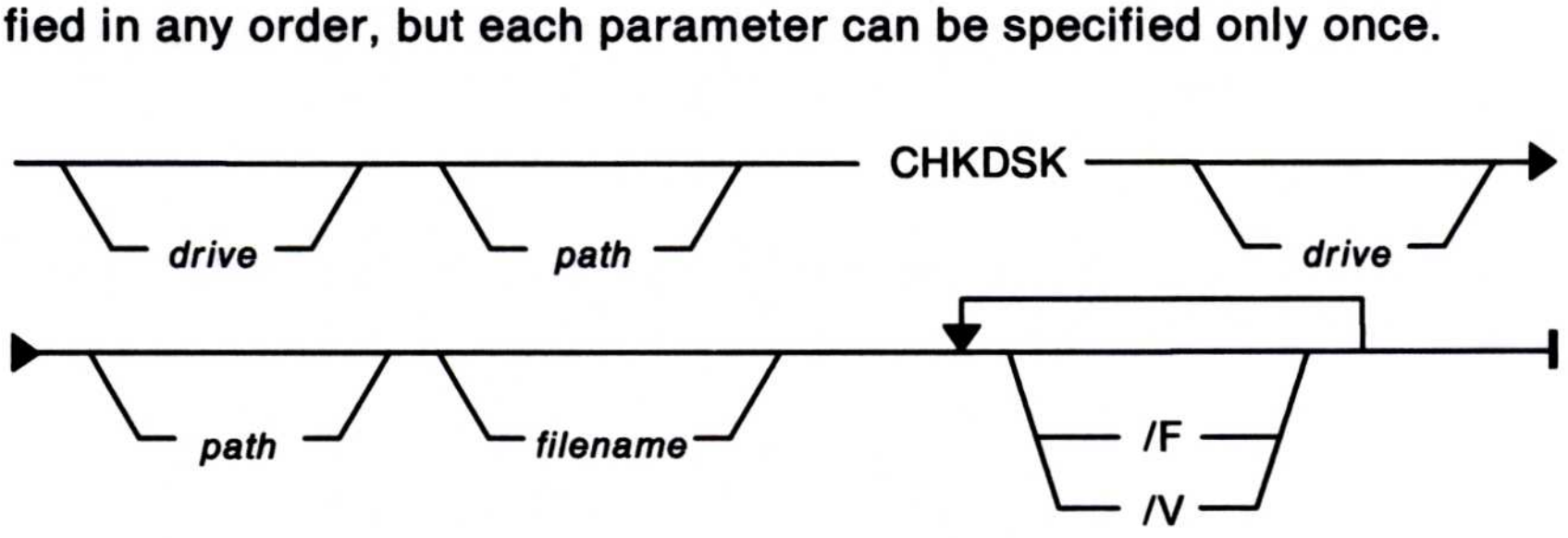
There are some commands that allow you to specify multiple filenames in OS/2 mode. You can enter up to 128 characters, including the command name, at the command prompt. Be sure you do not separate the parts of a file specification: drive, path, and filename. For example, you can enter:

```
TYPE C:\MYDIR\MYFILE.TXT A:\MEMO.TXT D:\REPORTS\APRIL
```

Note that in command syntax *filename* can consist of one to eight characters followed by an extension of one to three characters. The extension for the filename in the syntax diagram is optional, unless otherwise indicated in the **Parameters** list. When an extension is specified for a filename, it must be separated from the first part of the filename by a period, as in this example:

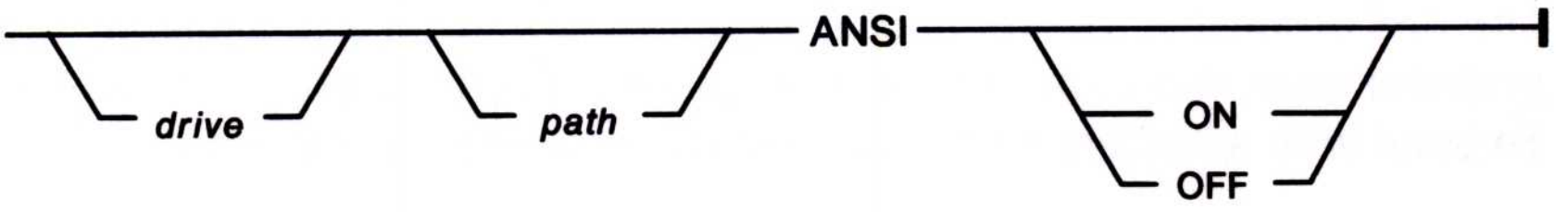
```
MYFILE.TXT
```

Specifying Items Once in Any Order: A stack of parameters with a return arrow means that one or more of the parameters can be specified in any order, but each parameter can be specified only once.



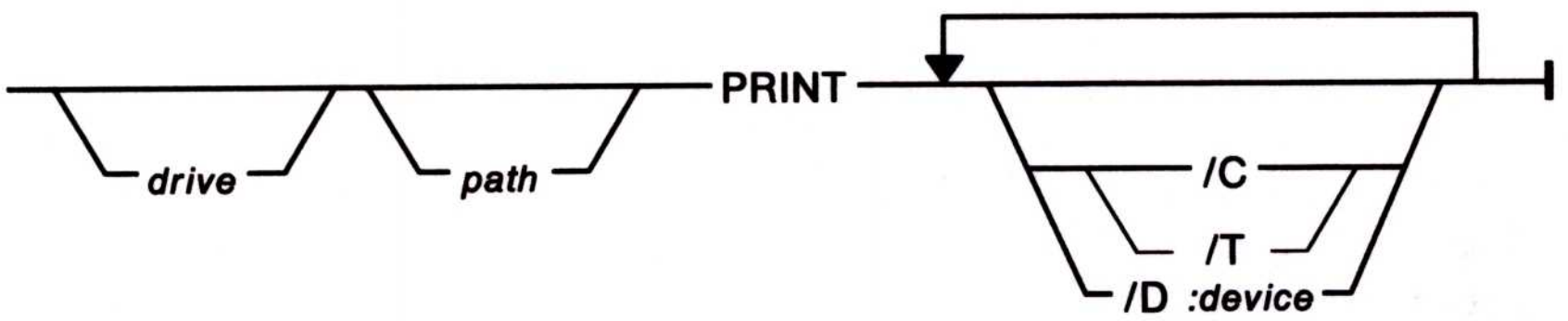
In this example if you want to specify both the /F and /V parameters, you can specify either one before the other.

Choosing One from a Stack: A stack of parameters with no return arrow means that you cannot choose more than one from the stack.



In this example, you can choose either ON or OFF.

Choosing One from within a Stack: A stack of parameters with no return arrow within a stack that has a return arrow means that you can choose only one from the smaller stack, in any order with the parameters in the larger stack, as in this example:



In this example you can specify /D:*device* and /C once in any order, or you can specify /D:*device* and /T once in any order, but you cannot specify /C and /T together.

Commands for Problem Determination

OS/2 has programs that monitor the operating system and assist in problem determination. These specialized programs are intended to be used with assistance from an IBM Service Representative.

The OS/2 commands for problem determination are:

CREATEDD
TRACE
TRACEFMT
TRACE statement in CONFIG.SYS
TRACEBUF statement in CONFIG.SYS.

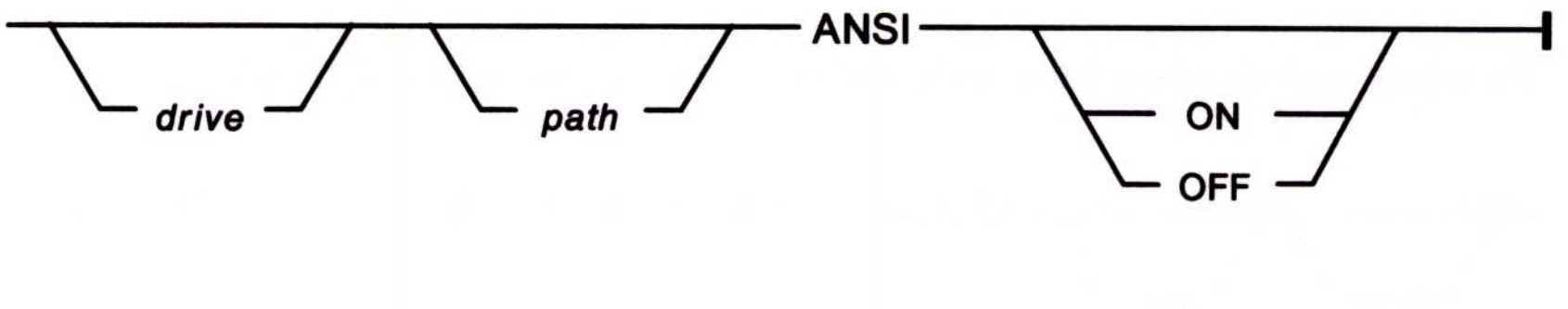
These commands are included in this book so that you can refer to them when talking to your IBM Service Representative.

ANSI (extended keyboard and display)

Purpose

Enables or disables extended display and keyboard support in OS/2 mode.

Syntax



Parameters

ON	Enables extended display and keyboard support. This is the default value.
OFF	Disables extended display and keyboard support.

Remarks

This command enables and disables ANSI control sequence processing in OS/2 mode. These functions allow you to redefine keys, manipulate the cursor, and change display color attributes. Entering this command without a parameter displays the current setting.

Example

To disable extended display and keyboard support in OS/2 mode, at the OS/2 command prompt enter:

```
ANSI OFF
```

Related Commands

DEVICE=ANSI.SYS in CONFIG.SYS, PROMPT

APPEND (search path)

Purpose

Sets a search path for data files outside the current directory.

Syntax

The first time APPEND is loaded

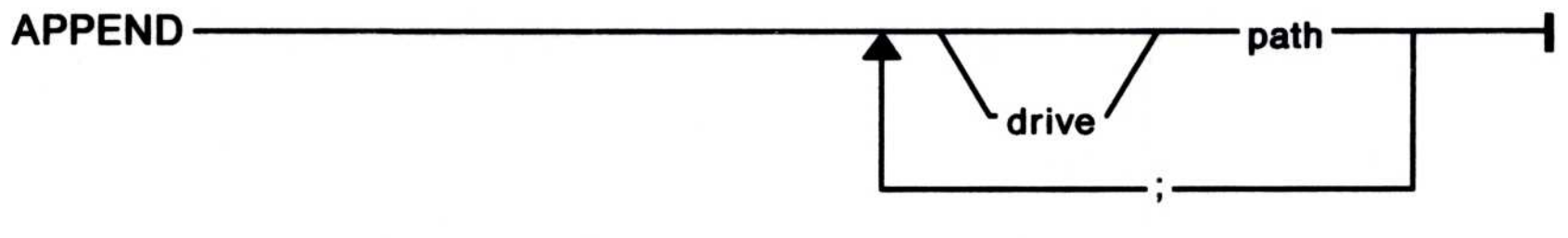
To keep paths specified with APPEND in the DOS environment:



Note: If you want to use the /E parameter to keep APPEND paths in the DOS environment, you must specify it the first time you use APPEND. You may not specify paths with it. For an explanation of the DOS environment, see the SET command on page 2-117.

At any time during the DOS session

To specify directories to be searched:



To delete usage of the APPEND command:



To display the list of appended directories:



Note: Although you may use any of the above syntaxes at any time during the DOS session, specifying either of the last two the first time you use APPEND would be of no value to you, since no paths have been specified yet with APPEND. These syntaxes show APPEND as already loaded. The first time you use APPEND, it is an external command; however, once loaded, it functions as an internal command. If, in the above syntaxes, you are using APPEND for the first time, you may need a drive and path to locate APPEND.

Remarks

If you specify the /E parameter the first time you use APPEND, then paths you specify with a subsequent APPEND command are kept in the DOS environment until you replace them with paths specified by another APPEND command. This is similar to the way the PATH command functions for paths to executable files. You may look at or modify APPEND paths in the environment by using APPEND; or you can use the SET command.

The SET command allows you to see all the values that are in effect for the DOS environment, including any paths specified by APPEND if the /E parameter has been specified first. You can also use SET to change or cancel any of these values. For more information on the SET command, see page 2-117.

The Advantage of /E: Some application programs query the environment. By using the /E parameter you allow these application programs (especially compilers and assemblers) to query the environment and, depending on what they find, change the actions they take.

When You Might Not Want to Use /E: If another command processor is started with the COMMAND command, it has the same environment as the previous command processor. However, if the path specified by APPEND is changed in the new command processor, the change is not made to the environment of the previous command processor. Thus, when you want to ensure that all processes are affected by changes to the APPEND path string, do not use the /E parameter.

If you specify a path with APPEND the first time you use it, this path is not stored in the environment and can only be viewed or changed with the APPEND command.

APPEND (search path)

If you have previously specified paths with APPEND and you enter:

```
APPEND
```

the current APPEND list is displayed.

A semicolon following the APPEND command resets the path to null:

```
APPEND ;
```

Note: Some programs read a file found in an appended directory and then write the file back to the disk. In these cases, the file is written to the *current directory*, leaving the original file unchanged.

You can append as many directories as you can specify in a total of 128 characters.

The search sequence for a specified file is:

1. Search the specified directory, or the current directory if the directory is not specified.
2. Search the directories indicated by the current APPEND command.

Why Use Append: APPEND is useful when you want to keep an application program and its associated data files in one directory and group information by category in other directories, such as \REPORTS, \MEMOS, and \LETTERS.

Examples

Suppose you are in your \REPORTS directory, and you want to use your word processing and your spread sheet programs and their respective data files. Your word processing application is located in the \WORDPROC directory, and your spread sheet program is located in the \CALC directory. If you enter the following:

```
PATH C:\WORDPROC;C:\CALC;
APPEND /E
APPEND C:\WORDPROC;C:\CALC;
```

your application programs can then query the environment for paths to data files. APPEND will also find any data files you specify that have these paths.

Suppose you decide not to use the /E parameter and instead enter:

```
PATH C:\WORDPROC;C:\CALC;
APPEND C:\WORDPROC;C:\CALC;
```

APPEND can still find any data files that are in these path locations, however the application programs cannot query the environment for paths to data files so that they can modify their behavior.

Note: Most application programs that need to query the environment for data files (for example, compilers and assemblers) will tell you this in the instructions that come with the program.

Related Commands

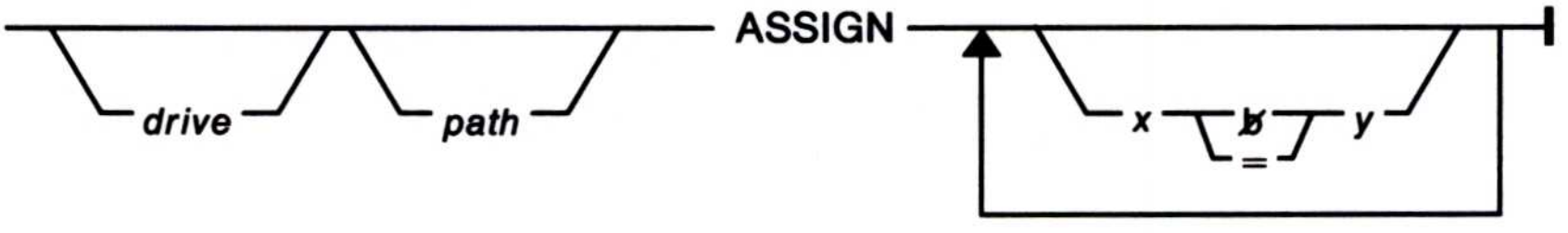
PATH, DPATH, SET

ASSIGN (assign drives)

Purpose

Assigns a drive letter to a different drive.

Syntax



Parameters

x	The drive to which reads and writes are currently sent. Do not type a colon after the drive letter.
y	The drive to which you want reads and writes sent. Do not type a colon after the drive letter.

Remarks

ASSIGN is an external command that allows you to perform disk operations on drives other than A and B for programs that only use those two drives. The command

```
ASSIGN A=C B=C
```

enables you to use drives other than A and B, such as a fixed disk (C). All references to drives A and B now go to drive C. You can use a space between the drive letters instead of the equal sign (=).

To reset all drives back to their original assignments, type: ASSIGN and press the Enter key.

You should not use the ASSIGN command with the BACKUP or PRINT commands. ASSIGN hides the true device type from commands that require actual drive information. The FORMAT and DISKCOPY programs ignore drive reassignments.

Examples

The following example assigns all requests for drive A to drive C.

```
ASSIGN A=C
```

Thus, if you enter:

```
DIR A:
```

OS/2 displays the directory that is on physical drive C.

The following example assigns any requests for drive A or drive B to drive C.

```
ASSIGN A=C B=C
```

The following example resets any previous drive assignments so that requests for drive A go to drive A.

```
ASSIGN
```

Related Commands

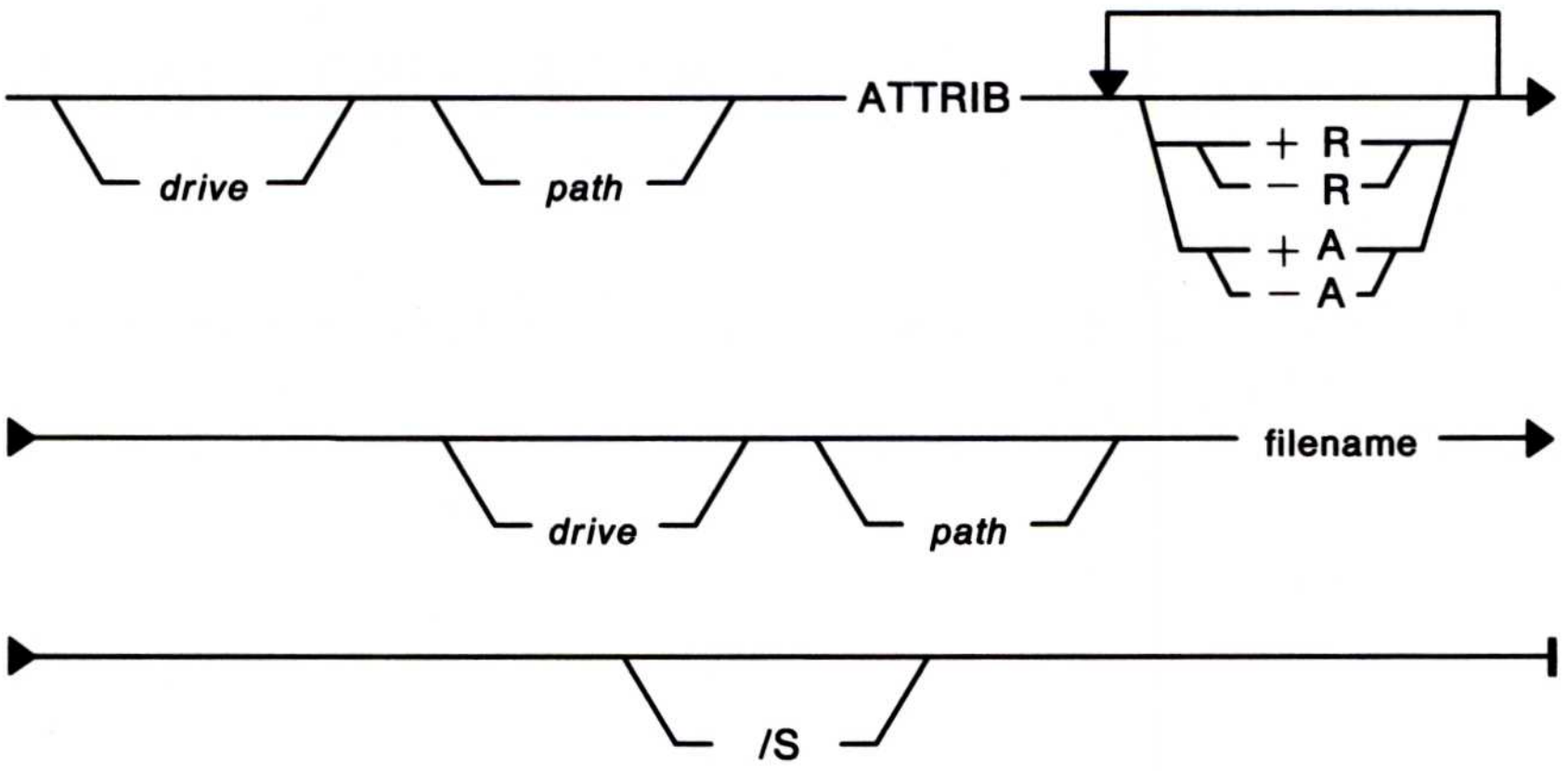
SUBST, JOIN

ATTRIB (file attribute)

Purpose

Turns on or off the read-only attribute and the archive bit of a file, for selected files in a directory or for all files in a directory.

Syntax



Parameters

+R	Turns on the read-only attribute of a file.
–R	Turns off the read-only attribute of a file.
+A	Turns on the archive bit of a file.
–A	Turns off the archive bit of a file.
/S	Processes all subdirectories.

Remarks

ATTRIB is an external command that is useful in keeping files from being modified. When the read-only attribute of a file is turned on, it can be read, but it cannot be written to and changed.

ATTRIB is also useful when you want to include a file or files in a selective copy done by BACKUP, RESTORE or XCOPY.

ATTRIB (file attribute)

When you create or modify a file, its archive bit is automatically turned on. The BACKUP, RESTORE, and XCOPY commands use these indicators to selectively copy only new and changed files. If you specify the /M parameter with BACKUP RESTORE, or XCOPY, they back up, restore, or copy only those files specified whose archive bits are turned on. Thus, if you want to include in this selective copy a file that has *not* changed, you can use ATTRIB to turn on its archive bit.

Examples

To turn on the read-only attribute and turn off the archive bit for a file named FILE1.TXT, enter:

```
ATTRIB +R -A FILE1.TXT
```

To display the current settings of the archive and read-only attributes for FILE1.TXT, enter:

```
ATTRIB FILE1.TXT
```

and the following will be displayed:

```
      R   C:\FILE1.TXT
```

which shows the read-only attribute is on and the archive bit is off.

To remove the read-only attribute from FILE1.TXT, enter:

```
ATTRIB -R FILE1.TXT
```

When you enter:

```
ATTRIB FILE1.TXT
```

to display the results, you see that the read-only attribute for FILE1.TXT is now off as well.

```
          C:\FILE1.TXT
```

ATTRIB (file attribute)

The following example sets the attribute of the file A:\PROG1.BAS to read-only.

```
ATTRIB +R A:\PROG1.BAS
```

To turn on the read-only attribute for files in subdirectories, you can enter:

```
ATTRIB +R C:\PROG.* /S
```

This command changes all files with the name PROG in all subdirectories of drive C to read-only.

Related Commands

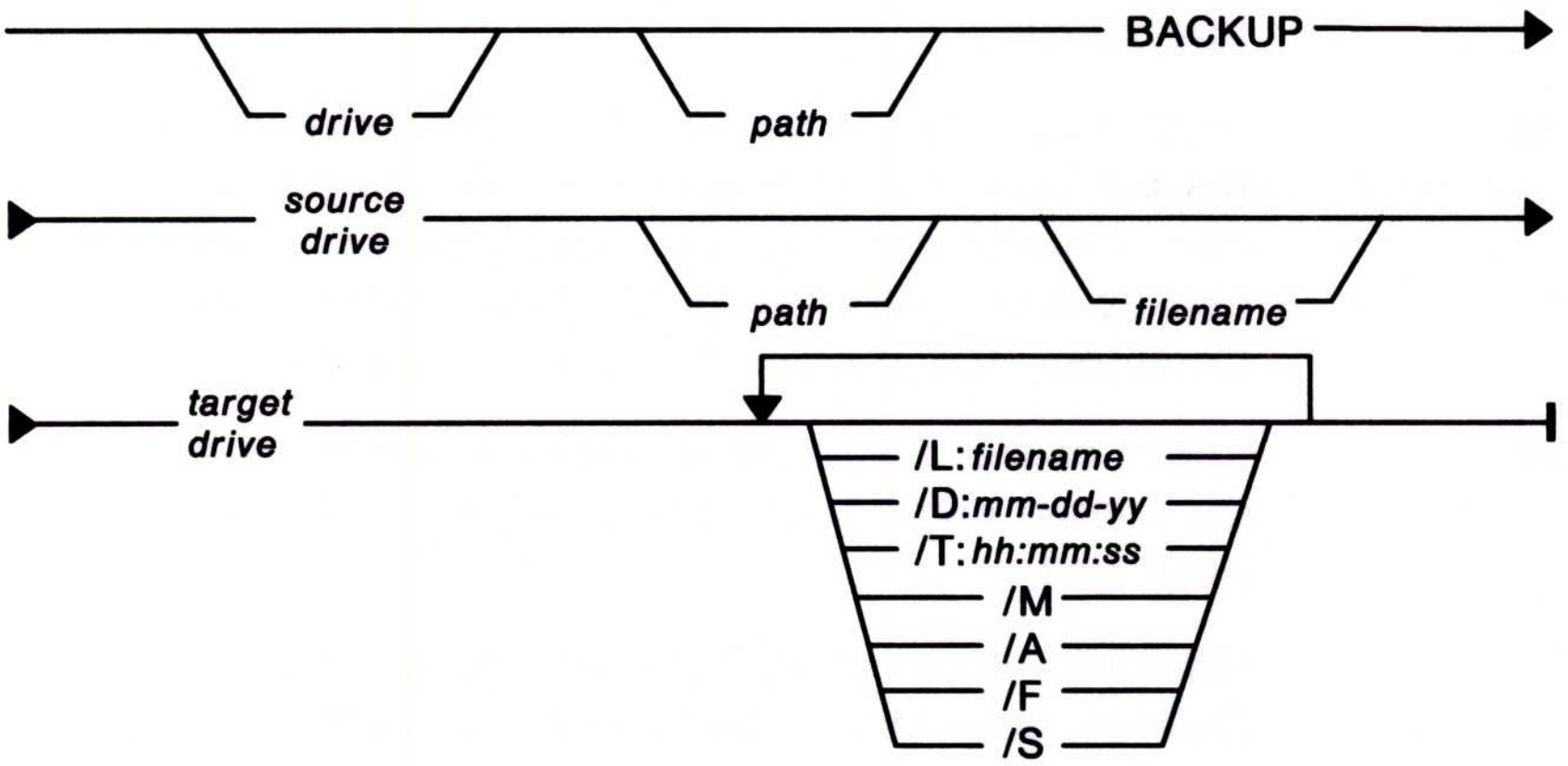
BACKUP, RESTORE, XCOPY

BACKUP (back up files)

Purpose

Backs up one or more files from one disk to another.

Syntax



Parameters

/L:filename Makes a backup log entry in the file specified. If no filename is given, the file BACKUP.LOG is placed in the root directory of the source drive. The first line of the entry in the file contains [date time] where date and time are the backup dates and times. Each subsequent line in the entry corresponds to one of the files that was backed up.

These lines consist of the filename and the number of the disk that contains the file. This information can be used when you need to restore a particular file from a diskette. You must specify which disk to restore so that BACKUP does not have to search for files. If the backup log file already exists, the current entry is appended to the file.

BACKUP (back up files)

/D:*mm-dd-yy* Backs up only those files that were last modified on or after the specified date. This parameter expects the date to be entered in the correct format for the current country specified with the COUNTRY command.

/T:*hh:mm:ss* Backs up only those files that were last modified at or after a certain time.

/M Backs up only those files that have changed since the last backup.

/A Adds the files to be backed up to those already on the backup disk. Does not erase old files on the backup disk. This parameter is not valid if files exist on the backup disk which were backed up by IBM Personal Computer DOS Version 3.2 BACKUP or earlier.

/F Specifies that the target diskettes are to be formatted by the FORMAT command. This parameter will not format a fixed disk.

Warning: When using the /F parameter, the target diskette capacity and the target drive capacity must be identical. For example, do not use a 360Kb diskette in a 1.2Mb diskette drive.

/S Backs up subdirectories also.

Remarks

This is an external command that can back up files on disks of different media types. For example, you can back up files from:

- Fixed disk to diskette
- Diskette to diskette
- Diskette to fixed disk
- Fixed disk to fixed disk.

BACKUP backs up files from one diskette to another, even if the disks have a different number of sides or sectors. The *source drive* is the name of the disk drive that you want to back up. You can add a path to specific files to be backed up. The *target drive* is the name of the backup disk drive. Unless otherwise specified, BACKUP erases the old files on a backup disk before adding new files to it.

BACKUP (back up files)

If you are backing up files from fixed disk to diskette, more than one diskette may be required.

Examples

This example backs up all files from the fixed disk drive C to the diskette in drive A.

```
BACKUP C:\*.*  A:/S
```

This example backs up file FILE.TXT from drive A to drive C.

```
BACKUP A:FILE.TXT C:
```

This example backs up all files from drive A that have been modified since the date 7−06−86 to drive C.

```
BACKUP A: C:/D:7-06-86
```

The following example adds the files on drive A to the backed up files already on the diskette in drive B.

```
BACKUP A: B:/A
```

Related Commands

ATTRIB, RESTORE, XCOPY

BREAK (set)

Purpose

Turns on or off the check for Ctrl+Break whenever a DOS program issues system calls.

Syntax



Parameters

ON Turns on the check for Ctrl+Break.

OFF Turns off the check for Ctrl+Break. If BREAK is not specified in CONFIG.SYS, this is the default value and means that OS/2 checks for Ctrl+Break being entered only during:

- Standard output operations
- Standard input operations
- Standard print operations
- Standard auxiliary operations.

Remarks

Some DOS programs may set themselves to respond to Ctrl+Break at any time. Setting BREAK will not affect these programs.

The ON parameter is useful to check for Ctrl+Break to exit a program or command that performs few or no screen or keyboard operations (such as a compiler).

Note: In OS/2 mode, the check for Ctrl+Break is always on. You cannot turn it off.

Stopping a Batch File's Processing: If a *filename*.BAT is running in DOS mode and you press Ctrl+Break, the following prompt is displayed:

```
Terminate batch job (Y/N)?
```

You can enter Y in response to this prompt, which causes the batch file to end and control to be returned to the operating system. You can also enter N, and the batch file will continue its processing.

However, when you press Ctrl+Break in OS/2 mode, you are not given this choice. Once you press Ctrl+Break to stop a *filename*.CMD from running, it ends and control is returned to the operating system.

Examples

The following example checks for Ctrl+Break only during screen, keyboard, and printer operations.

```
BREAK OFF
```

To find out what the current setting of BREAK is, type:

```
BREAK
```

Related Commands

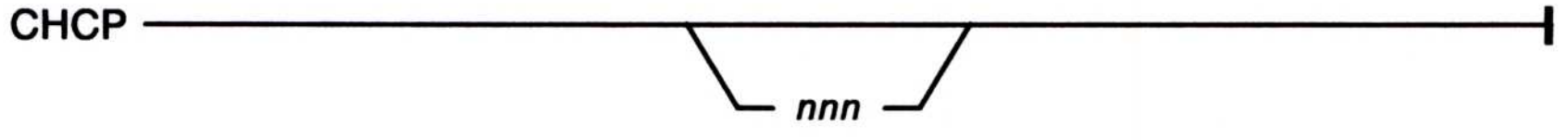
BREAK statement in CONFIG.SYS

CHCP (change code page)

Purpose

Allows you to switch back and forth between two code pages (defined character sets) that have been prepared for your system.

Syntax



Parameters

nnn Is the code page you want to use.

Remarks

The purpose of CHCP is to change the code page for the current session. Therefore, it is intended to be used at the command prompt. Any programs started under the *primary* command processor inherit the command processor's code page.

CHCP accepts either one of the two prepared system code pages as correct. If you enter the number of a code page that has not been prepared for the system, you receive an error message informing you of this.

To determine the code page being used, enter the following at either the OS/2 or DOS command prompt:

```
CHCP
```

If your system has been prepared for code page switching, you receive a message that tells you which code page is currently in use. If you enter CHCP at the OS/2 command prompt, you receive a message that looks like this:

```
Active code page:  437
Prepared system code pages:  437, 850
```

This tells you that code page 437 is active for the current OS/2 session, and that code pages 437 and 850 have been prepared for the system.

CHCP (change code page)

(When you enter CHCP in DOS mode, only the active code page is displayed.)

If no code pages have been prepared for your system, you receive a message telling you that code page support is not available on your system.

Why Switch Code Pages: If your system has been set up for code page switching, you can switch back and forth between two code pages:

- Your national language code page
- The multilingual code page.

Your national language code page contains the characters used by the language of your country and can include characters not found in any other code page. (The U.S. code page is IBM PC 437.) However, national language code pages are standard only for personal computers. Thus, their use is limited.

The multilingual code page (code page 850) is an international data processing standard. It has characters in common with each of the national language code pages and includes all characters for the languages of most European, North American, and South American countries. Since code page 850 is used by other IBM systems, data entered using these characters is transportable between the personal computer and other IBM systems. For example, suppose you create a file using characters from code page 850 and send it to someone who is using an IBM System/36. When that file is received and viewed or printed, it is identical to your copy, down to the last character. This means that identical operations performed on the data in the two files, such as searching and updating, produce identical results.

In most cases, OS/2 error messages use characters common to both your national language code page and the multilingual code page. So when you switch code pages, your OS/2 error messages remain the same.

CHCP (change code page)

Preparing for Code Page Switching: To successfully switch code pages, you have to include certain interrelated statements in your CONFIG.SYS file. These statements identify the two code pages to be used and also prepare devices. Not all devices are suitable for code page switching. For an explanation of the statements required for code page switching, see "Code Page Switching" on page 4-3.

To correctly set up your printer for code page switching, in addition to having the correct statements in your CONFIG.SYS file, you must also load the print spooler. If you wish to change code pages in DOS mode, you must start the print spooler before selecting the DOS command prompt from the Program Selector menu. See the SPOOL command on page 2-126 for information on how to start the print spooler.

If the printer is correctly set up for code page switching, printing jobs started in DOS mode or in the current OS/2 session after a successful CHCP is issued will print in the new code page.

Warning: If you specify a code page with the CHCP command that your printer has not been set up to use, a warning message like the following is displayed:

```
The code page "865" is not acceptable for the device.
```

If the printer has not been set up for code page switching, then no warning message is issued and the code page change will not be in effect for the printer.

If there is a DEVINFO statement for your printer and the print spooler is active for the printer, all output on that printer will default to the printer's Character Set 2. Using Character Set 2 is necessary to support printing in either of the two OS/2 code pages specified with the CODEPAGE statement in CONFIG.SYS. Character Set 1 can still be accessed by using escape sequences. Refer to the instructions that came with your printer.

Examples

If your system has been set up to use code pages 437 and 850, to change to the multilingual code page, enter:

```
CHCP 850
```

To change back to your national language code page, enter:

```
CHCP 437
```

Related Commands

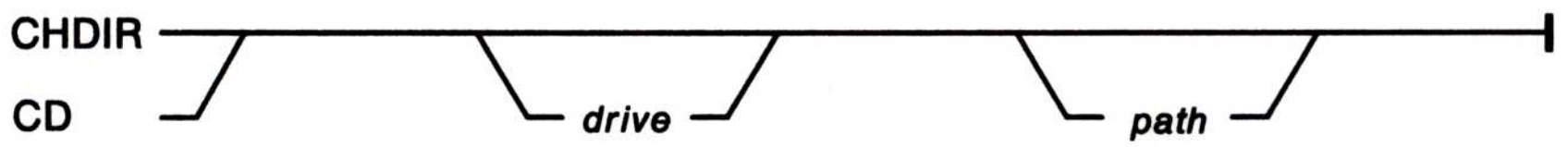
CODEPAGE, COUNTRY, and DEVINFO statements in CONFIG.SYS, SPOOL

CHDIR (change current directory)

Purpose

Changes the current directory or displays its name.

Syntax



Remarks

If you type CHDIR or CD without a path, it displays the name of your current directory. This is useful when you need to know the name of your current directory.

If the directory you wish to change to has a different path from your current directory, you must enter its entire path, including the root directory.

Examples

The following use of this command will put you in the *root* directory of your current drive:

```
CHDIR \
```

If you are in your root directory and you want to change to the directory called \OS2 you can enter:

```
CHDIR \OS2
```

When the directory you want to change to is directly under your current directory, you do not have to include the name of your current directory. In the example above, your current directory was the root directory. Thus, you did not have to type the leading backslash for the root directory. Suppose you are in the directory called \OS2. To get to the \OS2\INSTALL directory, you need only type:

```
CHDIR INSTALL
```

CHDIR (change current directory)

You can use ".." (the symbol for the parent directory) when you want to change to the parent directory of your current directory. For example, if you are in the C:\OS2\INSTALL directory and you want to change to the C:\OS2 directory, entering:

```
CHDIR ..
```

puts you in the parent directory of your current directory.

Related Commands

MKDIR, RMDIR, TREE

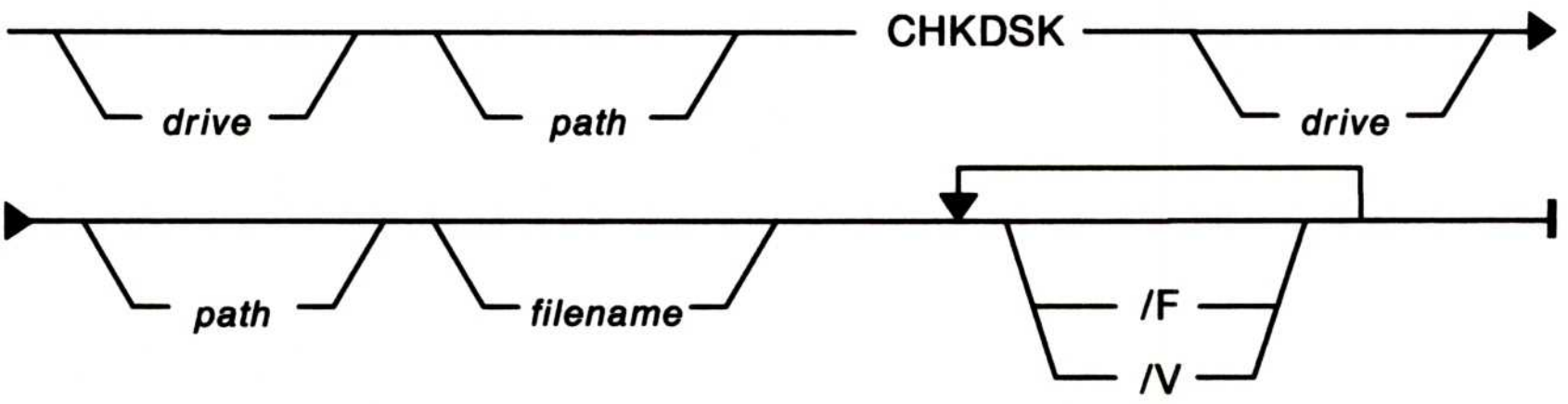
CHKDSK (check disk)

Purpose

Analyzes the directories, files, and the File Allocation Table on the specified or default drive and produces a disk status report.

Note: CHKDSK also produces a storage report for DOS mode only.

Syntax



Parameters

/F	Fixes errors found in your directory. Do not use this parameter if you redirect CHKDSK output.
/V	Displays all files and their paths on the default or specified drive.
filename	Displays the number of noncontiguous areas occupied by the file or files specified.

Remarks

This is an external command that should be run occasionally on each disk to check for errors. If any errors are found, CHKDSK will display the error messages and then a status report. When running in DOS mode, CHKDSK will display the storage report. If you type a filename after CHKDSK, OS/2 displays a status report for the disk and for the individual file.

You can redirect the output from CHKDSK to a file, if you want to save the status report for future use. Simply type:

```
CHKDSK C: > filename
```

The errors will be sent to the filename specified. Do not use the /F parameter if you redirect CHKDSK output.

To have CHKDSK fix a disk, specify the /F parameter. CHKDSK returns an error if there are any open files on the disk. While CHKDSK is fixing the disk, it prevents any other activity on it. Therefore the drive to be fixed cannot be the one from which you start OS/2, nor can it be the one that contains the CHKDSK program. In these instances, here is what you do:

- To fix the drive that is a fixed disk from which you normally start OS/2, insert a system diskette in diskette drive A. Start OS/2 and run CHKDSK in A, specifying C as the drive to be fixed.
- To fix the drive that contains CHKDSK, copy CHKDSK to another drive and run it from that drive by specifying the drive and path.

If the /F parameter is not specified and there are open files, CHKDSK may give a result that indicates there are lost clusters on the disk. This will occur if there are files open that have been written to but the file allocation table (FAT) has not been updated. If a large number of clusters are reported as lost, you should consider using the /F parameter to fix the disk.

DOS Mode Only: This command will not work on drives involved with the SUBST or JOIN command.

Examples

A sample status report for OS/2 mode looks like this:

```
Volume MYDISK created -- Jul 17, 1987  10:50am

 21317632 bytes total disk space.
   245760 bytes in 3 hidden files.
   120832 bytes in 56 directories.
 17244160 bytes in 1119 user files.
    30720 bytes in bad sectors.
  3676160 bytes available on disk.
```

CHKDSK (check disk)

A sample status report for DOS mode looks like this:

```
Volume MYDISK created -- Jul 17, 1987  10:50am

 21317632 bytes total disk space.
   245760 bytes in 3 hidden files.
   120832 bytes in 56 directories.
 17244160 bytes in 1119 user files.
    30720 bytes in bad sectors.
  3676160 bytes available on disk.

[DOS mode storage report ]
   655328 bytes total storage
   533136 bytes free
```

If *filename* is specified, CHKDSK displays the number of noncontiguous areas occupied by the file or files. Global filename characters are allowed. If you type:

```
CHKDSK C:FILE?.*
```

In addition to displaying the status report, CHKDSK also displays messages. For example, if FILE1, FILE3, and FILE6 contain noncontiguous blocks, CHKDSK displays the following:

```
C:\FILE1
    contains 2 non-contiguous blocks
C:\FILE3
    contains 2 non-contiguous blocks
C:\FILE6
    contains 4 non-contiguous blocks
```

Related Commands

RECOVER

CLS (clear screen)

Purpose

Clears the display screen.

Syntax



Remarks

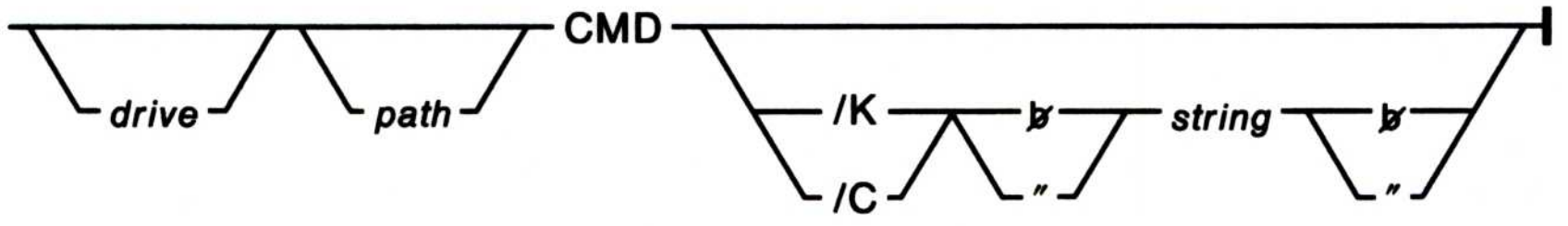
This command clears the display screen.

CMD (command processor)

Purpose

Starts another command processor in OS/2 mode.

Syntax



Parameters

/K string	Allows you to pass a command to CMD.EXE but does not automatically return to the previous command processor after the command is completed. The *string* is the command you want to pass.
/C string	Allows you to pass a command to CMD.EXE and automatically return to the previous command processor after the command is completed. The *string* is the command you want to pass.

Remarks

CMD.EXE is the OS/2 command processor.

Issuing CMD without any parameters causes another command processor to be started, which inherits the environment known to the previous command processor. For example, let's assume that the current OS/2 command processor uses the default OS/2 prompt. If you enter:

```
CMD
```

another command processor is started that inherits the default prompt.

If you change any of the environment variables in the current command processor, the change is known only to the current command processor. In this case, suppose you change the current

command processor's prompt to something other than the default prompt and then exit back to the previous command processor. The previous command processor will still have the default prompt.

Note: For a description of OS/2 environment variables, see page 2-117.

If the *string* specified with the /K or /C parameter contains command operators, then *string* must be enclosed in double quotes. For a description of OS/2 mode command operators, see Chapter 5, "Making the Command Processor Work for You" on page 5-1.

When another command processor is loaded either with the /K parameter or no parameters at all, you can return to the previous command processor by entering the EXIT command.

Example

When you use the /C parameter to pass a command to the command processor, the command is interpreted and acted upon as if you had typed it at the OS/2 prompt. For example, if you type:

```
CMD /C DIR B:
```

another command processor is loaded that processes the command DIR B: and returns you to the previous command processor.

Note: In OS/2, there is no limitation to the size of the environment, therefore a parameter to specify the size of the environment is not necessary.

Related Commands

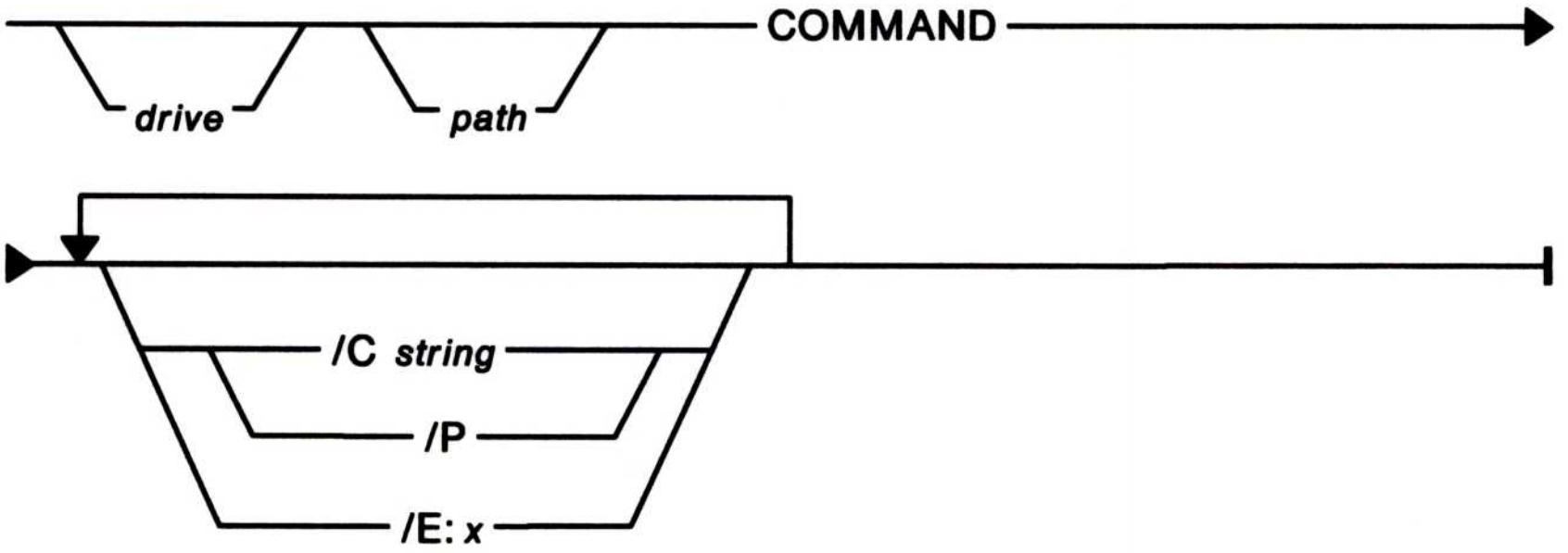
COMMAND, EXIT, START

COMMAND (command processor)

Purpose

Starts another DOS command processor.

Syntax



Parameters

/P Causes the DOS command processor you start to become permanent in storage. If you specify /P, the new command processor does not return to the previous command processor. You must restart OS/2 to remove the second DOS command processor.

/C string Allows you to pass a command to a secondary command processor and return to the primary command processor after the command is completed. The *string* is a command you want to pass to the command processor. The command is interpreted and acted upon as if you had entered it at the DOS command prompt. For example, if you type:

```
COMMAND /C DIR B:
```

a secondary DOS command processor is loaded, and it executes the command DIR B: and then exits back to the primary DOS command processor.

/E:X A number from 160 to 32768, specifying in number of bytes the size of the environment. This number is rounded up to a multiple of 16.

COMMAND (command processor)

Remarks

COMMAND.COM is the DOS command processor. If COMMAND.COM is not found in the specified directory, OS/2 then searches the environment for the value of COMSPEC. This system variable is placed in the environment when DOS mode is started. It describes the path the system uses to reload the command processor. You can change the value for COMSPEC with the SET command.

Issuing COMMAND without any parameters causes a new copy of the DOS command processor to be loaded, and this new copy will inherit the environment known to the previous level of the DOS command processor. If you use the SET command to change the environment known to the secondary DOS command processor, that change is known *only* to the secondary copy. Exiting back to the primary DOS command processor causes a resumption of the environment that the primary DOS command processor knew before the secondary copy existed.

Example

In this example, you want to start another DOS mode command processor, but you do not want it to return immediately to the previous command processor. You also want to specify the size of its environment as 16Kb. Enter the following at the DOS command prompt:

```
COMMAND /E:16000
```

When you are ready to return to the previous command processor, type:

```
EXIT
```

Related Commands

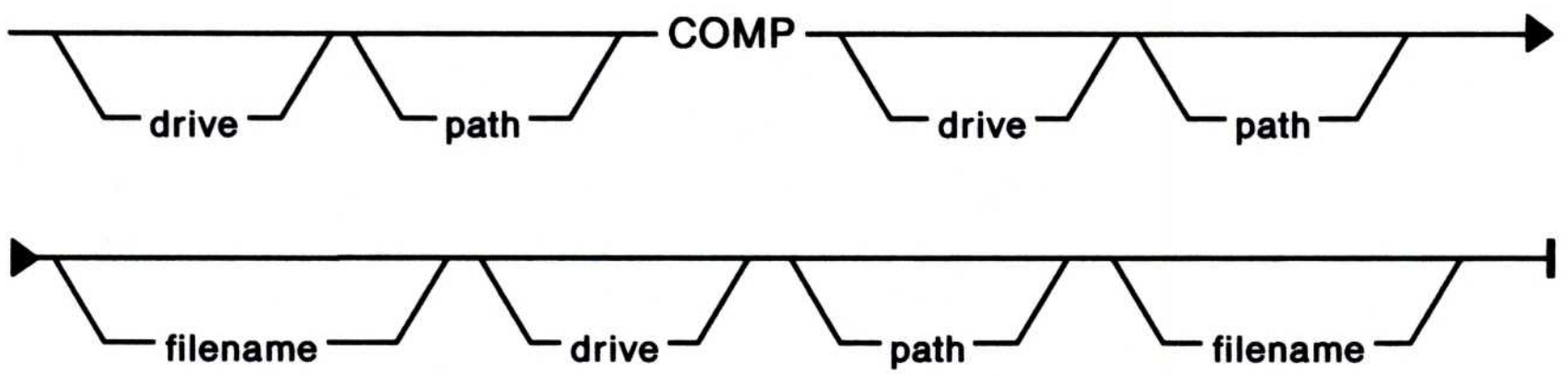
EXIT, CMD in OS/2 mode

COMP (compare files)

Purpose

Compares the contents of two files.

Syntax



Remarks

This is an external command. The files that you compare may be on the same drive or on different drives. They can also be in the same directory or different directories.

Global filename characters are allowed in both filenames, and cause all files matching the first filename to be compared with the corresponding files from the second filename. Thus, typing:

```
COMP A:*.ASM B:*.BAK
```

causes each file that has an extension of .ASM from the current directory of drive A to be compared with a file of the same name (but with an extension of .BAK) from the current directory of drive B.

If neither operand is specified with the COMP command, or if the second operand is missing, you are prompted for them. If either operand contains only a drive or a path with no filename, COMP assumes the filename is *.*. You can enter a complete path with either of the two filenames.

The paths and names of the files being compared are displayed as the comparing process proceeds. An error message will follow if a file matching the second filename cannot be found, or if a specified directory path is incorrect.

COMP (compare files)

It is possible to compare all files in one directory with all corresponding files in another directory.

If no file matches the first filename, you receive a message. For example, suppose you specify FILE1 as the first filename and no such file exists. You receive this message:

```
SYS1490: The system cannot find FILE1.

Do you want to compare more files (Y/N)?
```

During the comparison, an error message appears for any location that contains mismatching information in the two files. The message indicates the offset into the files of the mismatching bytes, and the contents of the bytes themselves (all in hexadecimal), as follows:

```
A COMPARE error occurred at OFFSET XXXXX
Mismatching byte of file 1 = XX

Mismatching byte of file 2 = XX
```

In this example, FILE 1 is the first filename typed; FILE 2 is the second filename typed.

After 10 unequal comparisons, no further comparing is done, and a message is displayed.

After a successful comparison, COMP displays a message informing you of this.

After the comparison of the two files ends, comparing will proceed with the next pair of files that match the two filenames, until no more files can be found that match the first parameter. Then COMP displays a message informing you that you have the option to compare two more files or end the comparison.

The two sets of files you want to compare can have the same path and filenames, provided they are on different drives.

If you only specify a drive for the second file, it is assumed that the second filename is the same as the first filename.

If the file sizes are different, COMP displays a message informing you of this and asks if you wish to continue. You now have the option to continue the comparison or to end the comparison. If you choose to

continue, the compare process will consist of the length of the smaller of the two files.

COMP does not wait for you to insert a diskette containing a file to be compared. Therefore, if a file to be compared is not on the same diskette as the COMP command itself, you should type COMP with no parameters. When COMP prompts for the filenames, you can insert the desired diskette and reply with the name of the file to be compared.

Examples

The following example compares all .ASM files on drive B with the files on drive C with the same filenames.

```
COMP B:*.ASM C:
```

The following example compares all files in the directory C:\LEVEL1 with corresponding files in the directory C:\LEVEL2.

```
COMP C:\LEVEL1 C:\LEVEL2
```

Related Commands

DISKCOMP

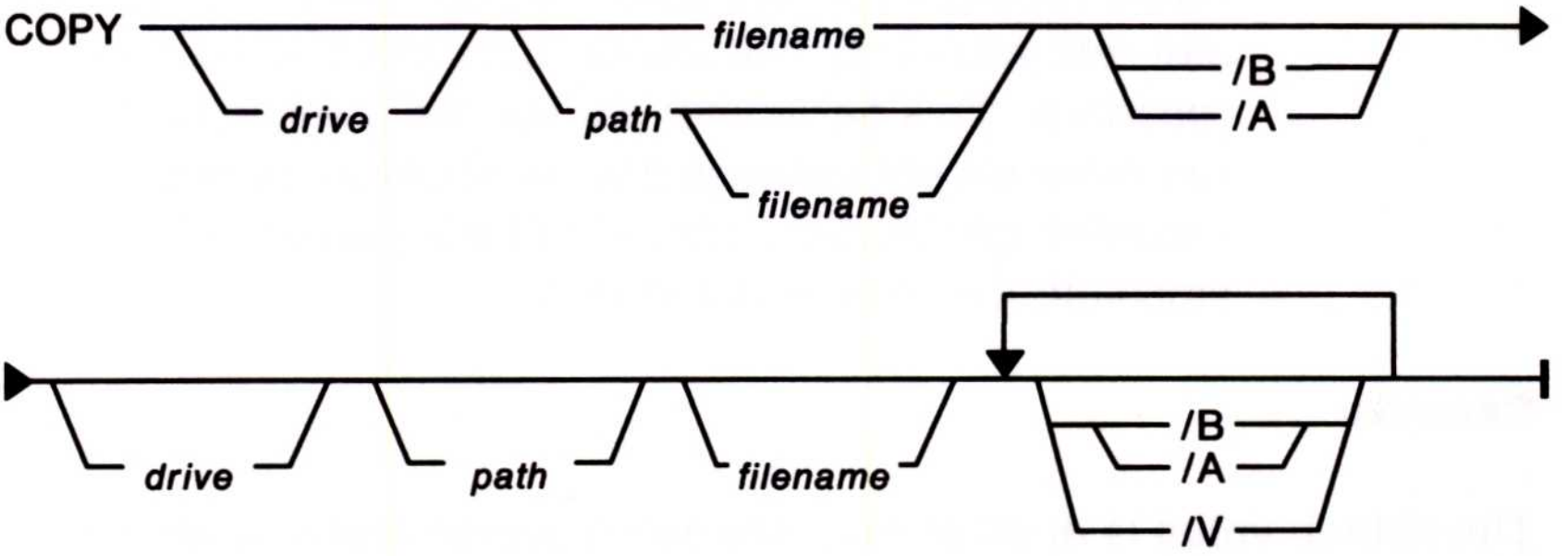
COPY (copy files)

Purpose

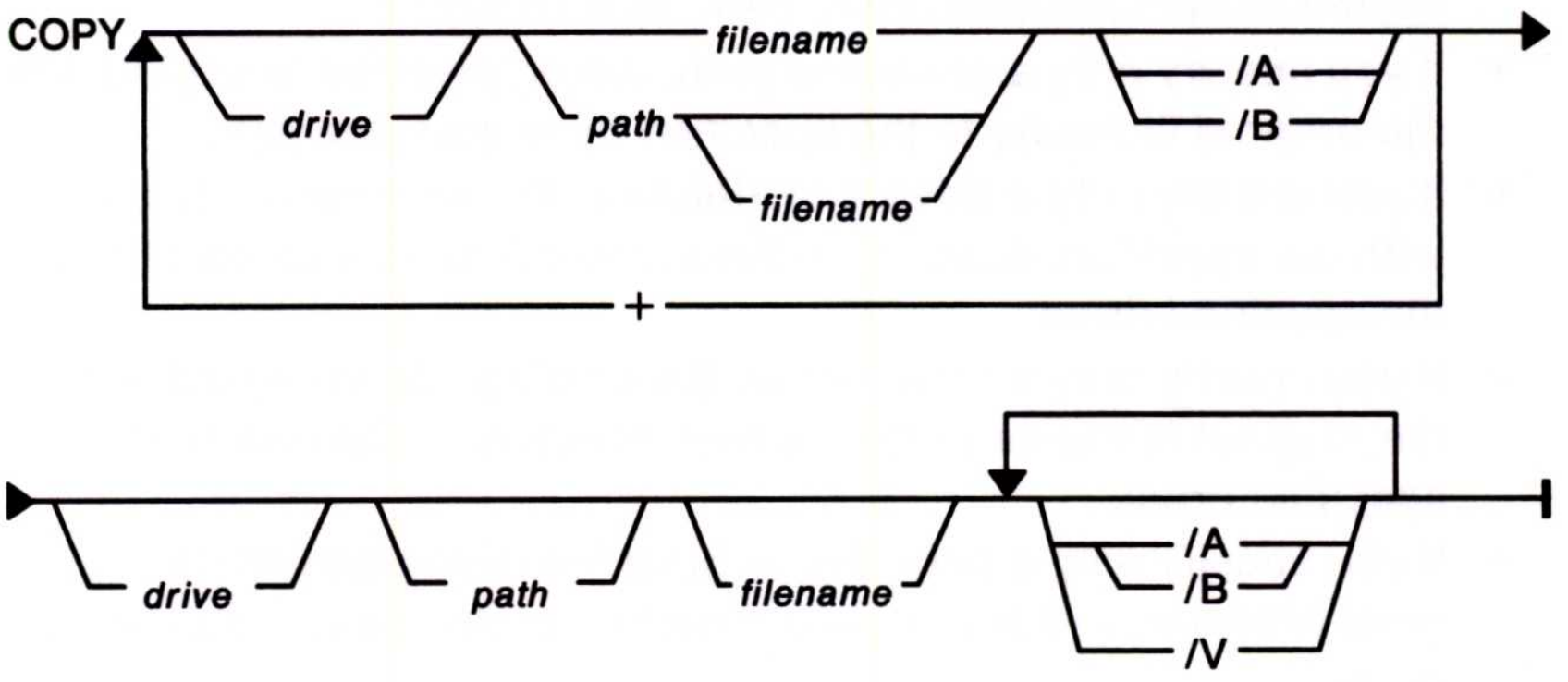
Copies one or more files. This command also combines files.

Syntax

To copy one or more files from the first operand to the second operand:



To combine files specified in the first operand into the file in the second operand:



Parameters

/A When used with a *source* filename, causes the file to be treated as an ASCII (text) file. Data in the file is copied up to but not including the first end-of-file mark (Ctrl+Z).

COPY (copy files)

When used with a *target* filename, causes an end-of-file character to be added as the last character of the file.

/B When used with a *source* filename, causes the entire file to be copied, including any end-of-file mark.

When used with a *target* filename, causes no end-of-file character to be added.

/V Causes OS/2 to verify that the sectors written on the target disk are recorded properly. Although there are rarely recording errors when you run COPY, this parameter lets you verify that critical data has been correctly recorded. This parameter causes the COPY command to run more slowly because OS/2 must check each entry recorded on the disk. OS/2 will display an error message if a write is not verified.

Remarks

The default value is /B when files are being copied and /A when files are being combined.

The second operand may take these forms:

- If you specify a drive, path, and filename, the original file is copied to the specified drive, path, and filename.
- If you specify only a drive and path, the original file is copied with the original filename to the specified drive and directory.
- If you specify only a drive and filename, the original file is copied with the specified filename to the current directory of the disk on the specified drive.
- If you specify only a drive name, the original file is copied with the original filename to the current directory of the disk in the specified drive.
- If you specify only a path, the original file is copied with the original filename to the specified directory of the disk in the default drive.
- If you specify only a filename, the original file is copied to the specified filename in the current directory of the disk in the default drive.
- If you do not specify a second operand, the copy will be made on the disk in the default drive and will have the same name as the

original file (first path). If the original file is on a disk in the default drive and if you do not specify the second path, the COPY command will stop and display an error message. (You are not allowed to copy a file to itself.)

Combining Files: The COPY command also allows you to combine files. Simply type any number of filenames as parameters to COPY, each filename separated with a plus sign (+); then specify a target filename into which you want the source files combined.

If one of the source files you want to combine has the same name as the target file, it will not be included in the combining process. COPY compares each source filename with the target filename. If the filenames do not match, the source file is appended to the target file. If they match, COPY displays an error message and does not append the file. Further file combining proceeds normally. This behavior of COPY lets you append files to an existing file. For example, suppose you enter:

```
COPY ONE.LST + *.LST
```

Since you have not specified a target filename, the default is ONE.LST. All files with an LST extension are appended to ONE.LST with the exception of ONE.LST itself. Although you receive an error message when COPY matches ONE.LST to the *.LST specification and doesn't append ONE.LST to itself, this is still a true physical append to ONE.LST.

Combining files is normally performed in text (ASCII) mode; that is, the first end-of-file character (Ctrl+Z) encountered is interpreted as the end of the file. When you use the /B parameter, this forces COPY to use the physical end-of-file (the file length shown when you use the DIR command).

You can combine ASCII files with binary files by using both the /A and /B parameters. You might want to use this feature of COPY when you need to incorporate text information placed after the first Ctrl+Z character of a binary file.

An /A or a /B parameter takes effect on the file it is placed after and applies to all subsequent files on the command line until the next /A or /B is encountered. If you specify a /B parameter for the target file, a Ctrl+Z character is added as the last character in the combined file. For example, suppose you enter:

COPY (copy files)

```
COPY FILE1.TXT + FILE2.TXT /B BIGFILE.TXT
```

The first file is treated as an ASCII file because the /A parameter is the default. The second file is treated as a binary file because of the /B specified. The target file does not get a Ctrl+Z added, because /A is the default.

Special Effects Using the Plus Symbol: As a rule, dates and times of files are changed only when you change their names or change their contents, and not when you merely copy them. However, you can use the plus symbol followed by two commas (+ ,,) with COPY to change the date and time of a file to the current date and time without altering its name or contents. For example, suppose you are in the current directory of default drive C and you enter:

```
COPY XYZ.ASM /B + ,, A:
```

The binary file XYZ.ASM in the current directory of C is copied to the current directory of A, and its date and time are changed. Using this feature of COPY in OS/2 mode, you can use global filename characters in the source file specification to change the dates and times of a group of files. For example, the command:

```
COPY A:\????BOOK.TXT + ,, C:\PUBS
```

copies all the files in the root directory of drive A that match the file specification to the \PUBS directory on drive C and changes their dates and times to the current settings for date and time. If you do not include a target file specification, for example:

```
COPY A:\????BOOK.TXT + ,,
```

then all files found that match the source file specification remain where they are, but their dates and times are changed. Note that when you use global filename characters with this feature in DOS mode, COPY finds only the first file that matches the file specification and changes its date and time.

Related Commands

DISKCOPY, RENAME, VERIFY, XCOPY

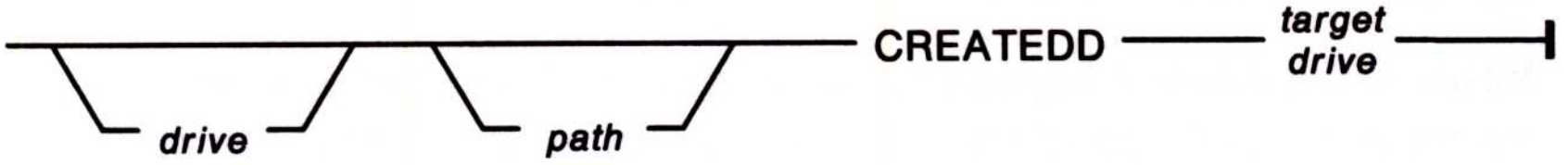
CREATEDD (create dump diskette)

Purpose

Creates a dump diskette.

Note: This command is intended for use with aid from an IBM Service Representative. For more information, see "Commands for Problem Determination" on page 2-6.

Syntax



Parameters

target drive Diskette drive that contains the diskette to be formatted as a dump diskette.

Remarks

This external command initializes a diskette so that it is usable by the stand-alone dump facility. The diskette is also made to appear full. This prevents you from accidentally placing data on the diskette. CREATEDD uses the FORMAT command. Therefore, FORMAT must be either in the current directory or along the path defined by the PATH command. Since the CREATEDD command creates only one diskette to *start* the storage dump, you should respond with "N" to the prompt "Format another?"

The Storage Dump: The stand-alone dump facility puts the contents of storage onto a set of diskettes that are often called *dump* or *customer service* diskettes. With the help of an IBM Service Representative, this facility can be used for problem determination following a system error that did not destroy the storage resident dump code and data.

The storage dump may span several diskettes; thus, you may want to have a supply of diskettes on hand. If additional diskettes are needed after the dump diskette created with CREATEDD has been used, you are

prompted for them. You can use formatted or unformatted diskettes; however if you have any data on formatted diskettes, it is overwritten.

Warning: You should use caution when starting a dump because all system activities in progress will stop without a cleanup (flushing buffers, and so on).

Starting the Dump: To start the storage dump, insert the dump diskette created with CREATEDD into drive A. Then, while holding down the Ctrl+Alt keys, press the Num Lock key twice. Note that although you have to be in OS/2 mode to *create* the dump diskette, you can start the actual dump in either mode.

Note: If the current keyboard translation table loaded with the KEYB command is not one of the OS/2 translation tables, it may not support the keyboard sequence required to make the dump work.

Once you have started the storage dump, the current contents of storage is then dumped to diskette. The stand-alone dump facility begins with address zero and copies storage contiguously until all of storage has been placed on diskette. This includes the 640Kb to 1Mb address area. If your computer does not have certain storage options installed (for example 512Kb to 640Kb), a special compression character is copied to diskette to indicate that this storage is not installed on your computer. This means that the smallest storage dump performed contains 1Mb of storage.

Ending the Dump: You may end the storage dump whenever you are prompted to insert another diskette by reinserting the dump diskette that you used to start the dump. (This is the diskette that you created with CREATEDD.)

Example

To create a dump diskette in diskette drive A, insert a diskette in drive A and enter:

```
CREATEDD A:
```

Related Commands

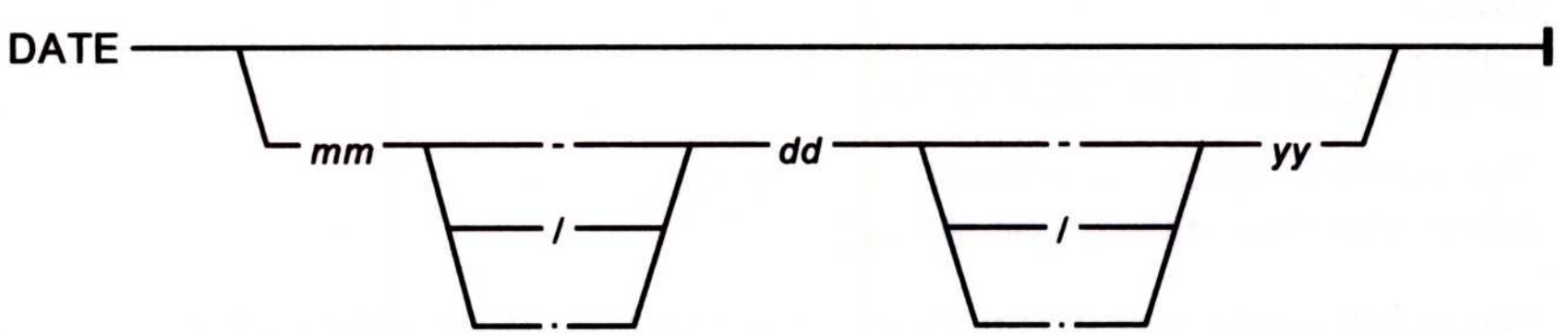
TRACE and TRACEBUF statements in CONFIG.SYS, TRACE, TRACEFMT

DATE (set system date)

Purpose

Displays or changes the date known to the system and resets the date on your computer's clock. This date is recorded in the directory when you create or change a file.

Syntax



Parameters

mm The new month must be entered using only the numbers 1 through 12.

dd The new day must be entered using only the numbers 1 through 31.

yy The new year must be entered using only the numbers 1980 through 2079, or the numbers 00 through 99. The 00–79 range represents the years 2000 through 2079, and the 80–99 range represents the years 1980 through 1999.

Remarks

OS/2 is programmed to change months and years correctly, whether the month has 31, 30, 29, or 28 days.

You can change the date at the command prompt or from a batch file. OS/2 does not automatically prompt you for the date when the OS/2 command processor starts. When the DOS command processor starts, you are prompted for the date only if no AUTOEXEC.BAT file exists.

It is possible to change the *mm-dd-yy* format in which the date is displayed and entered. The COUNTRY command in the CONFIG.SYS file

DATE (set system date)

allows you to do this. For example you can change to the French format *dd-mm-yy*, or to the Japanese format *yy-mm-dd*.

This command expects the date to be entered in the correct format for the current country specified with the COUNTRY command.

Example

If you enter:

```
DATE
```

DATE responds with the message:

```
The current date is weekday mm-dd-yyyy
Enter the new date: (mm-dd-yy):_
```

Press the Enter key if you do not want to change the date shown. You can also type a specific date after the DATE command, as in:

```
DATE 7-17-87
```

In this case, the "Enter the new date:" prompt does not appear after you have pressed Enter.

Related Commands

COUNTRY statement for CONFIG.SYS, TIME

DETACH (process noninteractively)

Purpose

Starts a noninteractive program.

Syntax



Parameters

command Any program or external OS/2 command and its options that does not require any input from the keyboard or mouse, or output to the display.

Remarks

The detached process should not issue any input or output calls to the keyboard, mouse, or display.

The DETACH command starts and simultaneously detaches an OS/2 program from its command processor. Therefore, any program that is started with DETACH must be able to process independently outside the control of the command processor.

The DETACH command can be used with redirection sequences to redirect a program's standard input and output to devices other than the keyboard and the display, allowing the program to then process noninteractively. For an explanation and examples of how this is done, see "Running Noninteractive Programs" on page 5-9.

DETACH (process noninteractively)

Example

Assuming you have created your spool directory, to start the print spooler and have it process as a noninteractive session, enter:

```
DETACH SPOOL
```

For more information on the SPOOL command, see page 2-126.

Related Commands

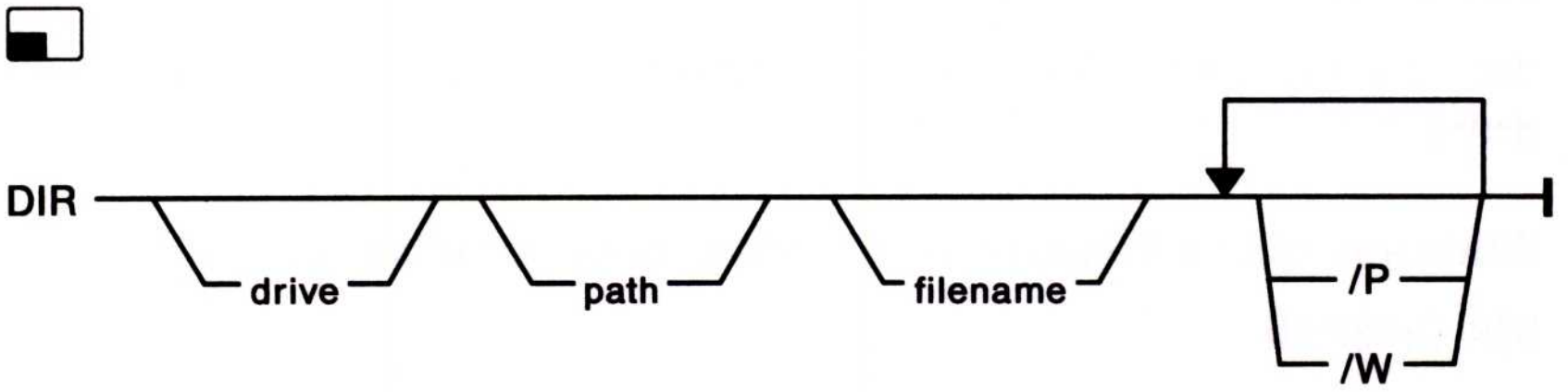
RUN statement in CONFIG.SYS

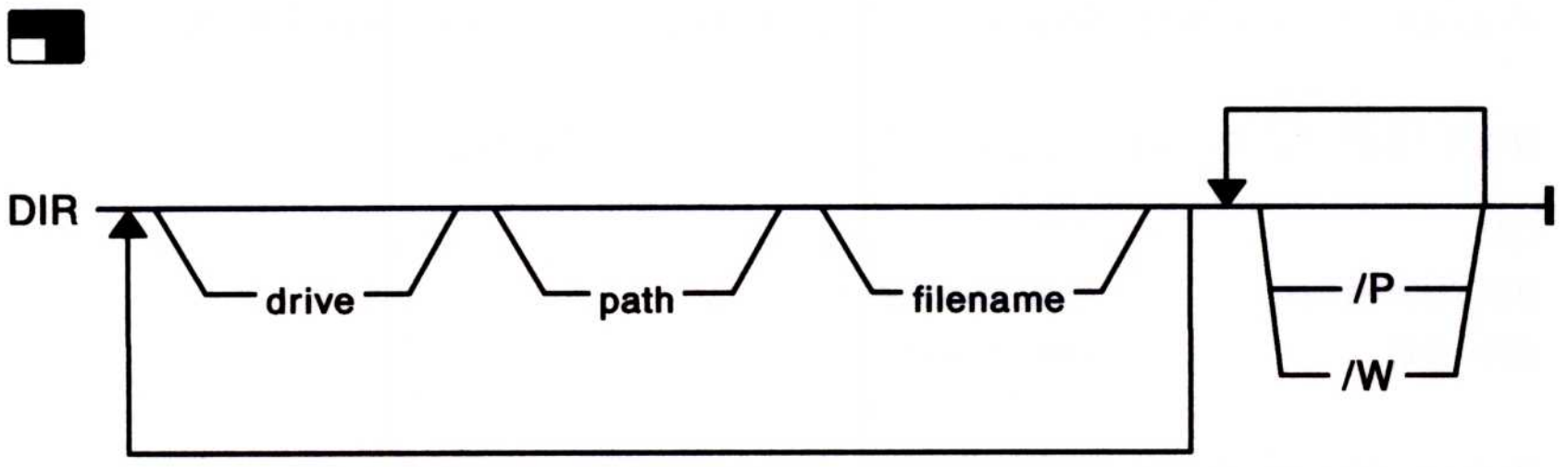
DIR (display files in directory)

Purpose

Lists the files in a directory.

Syntax





Parameters

/P Causes display of directory to pause after the screen is filled. To continue display of the output, press any key.

/W Selects wide display. In DOS mode, five filenames per line are shown without other file information. In OS/2 mode, five names are shown for 80-character width display mode, and two names are shown for 40-character width display mode.

DIR (display files in directory)

Remarks

This command is useful for finding out what is on a disk. If you just type DIR, all directory entries in the current directory on the default drive are listed. If you include a drive name, such as DIR C:, DIR lists all entries on the disk in the specified drive. If you type the filename without an extension, for example:

```
DIR REPORT
```

then DIR lists all the files named REPORT on the disk in the default drive.

When you type a filename with a drive letter, for example:

```
DIR C:REPORT
```

DIR displays the volume label, the directory name, the file information for REPORT, and the free space on drive C.

You can use global filename characters (? and *) in the filename.

Note that the following pairs of DIR commands are equivalent:

DIR	DIR *.*
DIR *filename*	DIR *filename*.*
DIR *.ext*	DIR *.ext

In OS/2 mode, if you enter more than one filename, for example:

```
DIR REPORT MEMO
```

then DIR displays the volume label of the default drive, the name of the default directory, the file information for the files REPORT and MEMO, and the amount of free space on the default disk.

When you specify multiple drives and paths for filenames in OS/2 mode, DIR displays:

- The volume label for each drive listed
- The free space for each drive listed
- The directory name for each file matching the file specification
- The file information.

In all cases, files are listed with their size in bytes and with the time and date of their last modification.

DIR (display files in directory)

Note: If the COUNTRY command in the CONFIG.SYS file is set to a country other than the U.S., the directory date and time formats may differ.

Example

The following example lists all the directory entries on the default drive.

```
DIR
```

The following example lists all the directory entries for the current directory of drive C.

```
DIR C:
```

The following example lists all the directory entries for the directory path \LEVEL2:

```
DIR \LEVEL2
```

If you wish to see the files in the parent directory of your current directory, you can enter:

```
DIR ..
```

The following example lists the directory entry of the file named FILE3.A in the current directory of the default drive.

```
DIR FILE3.A
```

The screen may look like this:

```
The volume label in drive C is MYDISK.
 Directory of  C:\

FILE3       A      1613     7-17-87   12:14p
     1 Files    141312 bytes free
```

If you enter:

```
DIR *.A
```

the screen may look like this:

```
The volume label in drive C is MYDISK.
 Directory of  C:\

FILE1       A     10368      7-20-87   12:13p
FILE3       A      1613      7-17-87   12:14p
     2 Files    141312 bytes free
```

DIR (display files in directory)

If you enter:

```
DIR FILE1
```

the screen may look like this:

```
The volume label in drive C is MYDISK.
 Directory of  C:\

FILE1       A    10368     7-20-87  12:13p
FILE1             2288     7-02-87  12:25p
     2 Files    141312 bytes free
```

When you do not specify an extension, the default for extension is all (*).

To display only the entries for files that have no extensions, enter the filename followed by a period. The .*ext* does *not* default to *. Using the above example, if you enter:

```
DIR FILE1.
```

only the file without an extension is displayed.

Related Commands

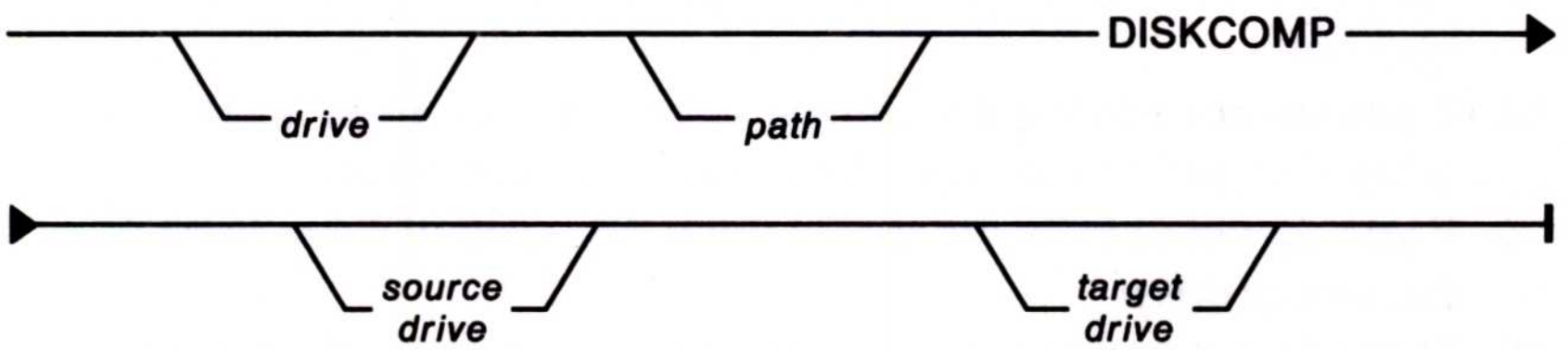
COUNTRY statement in CONFIG.SYS

DISKCOMP (compare diskettes)

Purpose

Compares the contents of the diskette in the source drive to the contents of the diskette in the target drive.

Syntax



Remarks

DISKCOMP is an external command. Drive specifications may be different or the same. If the same, a one-drive comparison is performed. You are prompted when to insert the appropriate diskette. After prompting, DISKCOMP waits until the Enter key is pressed before proceeding. Because of the OS/2 multitasking operating environment, DISKCOMP locks the diskettes to prevent any other program from reading or writing to the diskettes while the comparison is being performed.

DISKCOMP compares all tracks on a track-for-track basis, issuing a message if the tracks are not equal. The message indicates track number and side (0 or 1) where the mismatch occurred.

After the comparison, DISKCOMP prompts you for more comparisons with a message. If you type Y, the next comparison is done on the same drives originally specified. To end the command, type N.

DISKCOMP (compare diskettes)

DOS Mode Only: DISKCOMP does not recognize drive reassignments made by ASSIGN or SUBST commands.

Notes:

1. DISKCOMP determines the number of sides and sectors per track to be compared based on the source drive and diskette.
2. DISKCOMP does not require any additional parameters to determine media type; therefore any such parameters specified, such as the IBM Personal Computer DOS parameters **/1** or **/8**, are ignored.
3. If you do not specify the source drive and the target drive, a one-drive comparison is performed on the default drive.
4. If you do not specify the target drive, the default drive is used as the target drive.
5. If you do not specify the target drive and you specify the default drive as the source drive, a one-drive comparison is performed on the default drive.
6. DISKCOMP produces a message if a diskette error occurs. The error indicates the side and track of the failure. DISKCOMP continues to compare the remainder of the diskette but produces an error message at the end.
7. Neither the source drive nor the target drive can be fixed disks or virtual drives.

Diskette and Drive Compatibility: Listed here are the allowable DISKCOMP combinations.

Double-sided diskette drives

- 5.25 inch diskettes:
  - Single-sided to single-sided
  - Double-sided 320/360Kb to double-sided 320/360Kb
- 3.5 inch diskettes:
  - Double-density 720Kb to double-density 720Kb

DISKCOMP (compare diskettes)

High capacity diskette drives

- 5.25 inch diskettes:
  - Single-side to single-sided
  - Double-sided 320/360Kb to double-sided 320/360Kb
  - High capacity 1.2Mb to high capacity 1.2Mb
- 3.5 inch diskettes:
  - Double-density 720Kb to double-density 720Kb
  - High density 1.44Mb to high density 1.44Kb

High capacity diskette drive to double-sided diskette drive

- 5.25 inch diskettes:
  - Single-sided to single-sided
  - Double-sided 320/360Kb to double-sided 320/360Kb

No other combinations are allowed. In most cases, if an incorrect combination is specified, an error message tells you that drive types or diskette types are not compatible.

Example

The following example compares the diskette in drive A to the diskette in drive B.

```
DISKCOMP A: B:
```

Related Commands

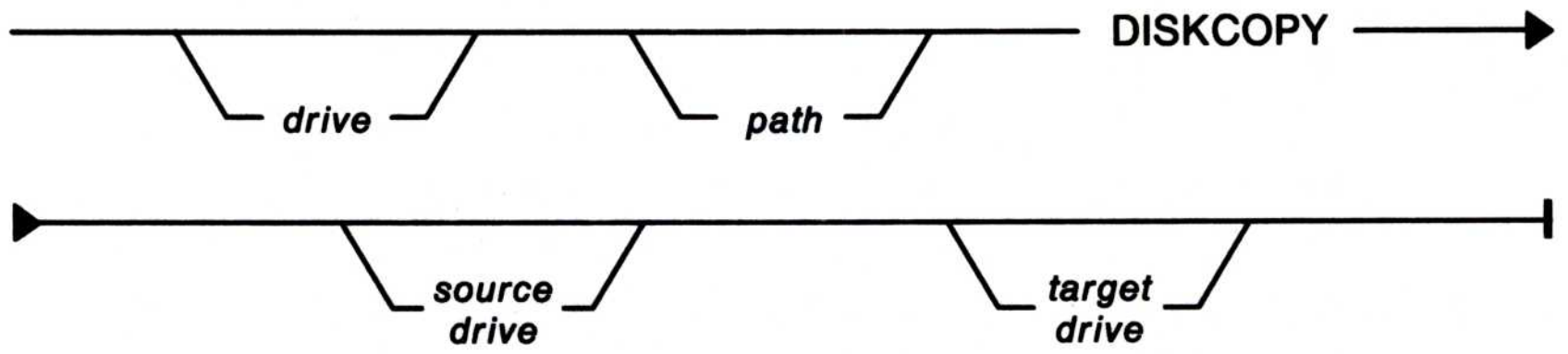
COMP, DISKCOPY

DISKCOPY (copy diskettes)

Purpose

Copies the contents of the diskette in the source drive to the diskette in the target drive. The target diskette is formatted, if necessary, during the copy.

Syntax



Remarks

DISKCOPY is an external command. Drive specifications may be different or the same. If the same, a one-drive copy operation is performed. You are prompted when to insert the appropriate diskette. After prompting, DISKCOPY waits until the Enter key is pressed before proceeding. After copying, DISKCOPY prompts you for more copies with a message. Because of the OS/2 multitasking operating environment, DISKCOPY locks the diskettes to prevent any other program from reading or writing to the diskettes while the copy is being performed.

DOS Mode Only: DISKCOPY does not recognize drive reassignments made by ASSIGN or SUBST commands.

Notes:

1. DISKCOPY determines the number of sides and sectors per track to be copied based on the source drive and diskette.
2. DISKCOPY does not require any additional parameters to determine media type; therefore any parameters specified, such as the IBM Personal Computer DOS parameters **/1** or **/8**, are ignored.
3. The target diskette created is an identical copy of the source diskette except for the volume serial number.

4. If the target diskette is unformatted, DISKCOPY formats it with the same number of sides and sectors per track as the source diskette.
5. If you do not specify the source drive and the target drive, a one-drive copy operation is performed on the default drive.
6. If you do not specify the target drive, the default drive is used as the target drive.
7. If you do not specify the target drive and you specify the default drive as the source drive, a one-drive copy operation is performed on the default drive.
8. If errors are encountered on either diskette during the copy operation, DISKCOPY indicates the drive, track, and side where it found the error and proceeds with the copy.
9. Neither the source drive nor the target drive can be fixed disks or virtual drives.

Diskette and Drive Compatibility: Listed below are the allowable DISKCOPY combinations.

Double-sided diskette drive to double-sided diskette drive

- 5.25 inch diskettes:
  - Single-sided to single-sided
  - Double-sided 320/360Kb to double-sided 320/360Kb
- 3.5 inch diskettes:
  - Double-density 720Kb to double-density 720Kb

High capacity diskette drive to high capacity diskette drive

- 5.25 inch diskettes:
  - Single-sided to single-sided
  - Double-sided 320/360Kb to double-sided 320/360Kb
  - High capacity 1.2Mb to 1.2Mb
- 3.5 inch diskettes:
  - Double-density 720Kb to double-density 720Kb
  - High density 1.44Mb to high density 1.44Mb

High capacity diskette drive to double-sided diskette drive

- 5.25 inch diskettes:
  - Single-sided to single-sided
  - Double-sided 320/360Kb to double-sided 320/360Kb

No other combinations are allowed. In most cases, if an incorrect combination is specified, an error message tells you that drive types or diskette types are not compatible. However, here are some instances where you may or may not get an error message:

- If the target diskette is of a different type than the source diskette but is *unformatted*, then the "incompatible" message does not occur. DISKCOPY attempts to format the target diskette according to the source's type, may or may not report errors in the copy process, and may or may not produce a valid copy.
- In the case where the target diskette *is* formatted, but to a higher capacity than the drive it is in, the drive cannot recognize it as formatted. Thus, DISKCOPY attempts to format it, with unreliable results.

Example

The following example copies the diskette in drive A to the diskette in drive B.

```
DISKCOPY A: B:
```

The following example copies the source diskette in drive A to the target diskette in drive A. You must switch diskettes as prompted.

```
DISKCOPY A: A:
```

Note: If the system was started from drive A, the system prevents you from using DISKCOPY to copy the system startup diskette from drive A. Therefore, to make a DISKCOPY of the system startup diskette, insert it in drive B and the blank diskette in drive A and enter:

```
DISKCOPY B: A:
```

Related Commands

COPY, XCOPY, DISKCOMP

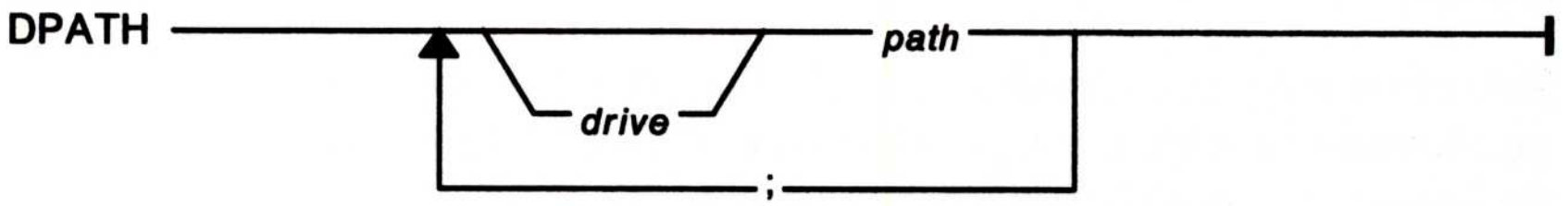
DPATH (search path)

Purpose

Gives application programs the search path to data files outside the current directory.

Syntax

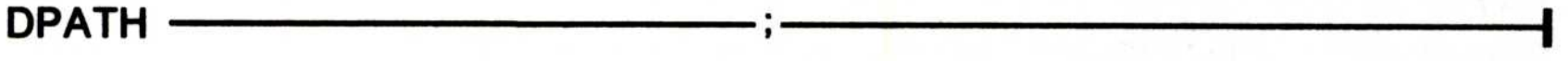
To specify paths to data files outside the current directory:



To display paths to data files currently in effect:



To delete usage of the DPATH command:



Remarks

This internal command allows you to tell application programs which directories to search for their data files, provided the application program makes use of the DPATH directory list.

DPATH is a system environment variable, which means application programs can query the environment for its value, and, depending on what they find, change their behavior. For more information on the OS/2 environment, see the SET command on page 2-117.

Like the PATH command, the number of directories you can specify with DPATH is limited only by the length of the command line, which cannot exceed 128 characters. Also like PATH, DPATH is an internal command.

DPATH (search path)

Once you set a search path for data files with DPATH, the path remains in effect for the current command processor until you replace it with another DPATH command.

To determine the current value of DPATH, enter:

```
DPATH
```

The default value for DPATH is no path. If you have specified a path with DPATH and decide you want no path specified, you can reset the path to null by entering:

```
DPATH ;
```

System installation creates a batch file called OS2INIT.CMD, which runs each time you select the OS/2 command prompt from the Program Selector. It contains this DPATH command:

```
DPATH C:\;C:\OS2;C:\OS2\INSTALL;
```

For more information on this special batch file, see "Special Batch Files" on page 3-4.

Example

Suppose you want to be able to find data files in the \LETTERS and \REPORTS directories on your C drive. A DPATH command is already specified in your OS2INIT.CMD file and is in effect for the current session. The existing DPATH command looks like this:

```
DPATH C:\;C:\OS2;C:\OS2\INSTALL;
```

To append the paths of the \LETTERS and REPORTS directories to the DPATH command already in effect, you can enter:

```
DPATH %DPATH%C:\LETTERS;C:\REPORTS;
```

In this example, the current path to data files is entered as a replaceable parameter. Note that it is assumed your current path ends with a semicolon. For more information on replaceable parameters, see the SET command on page 2-117.

Related Commands

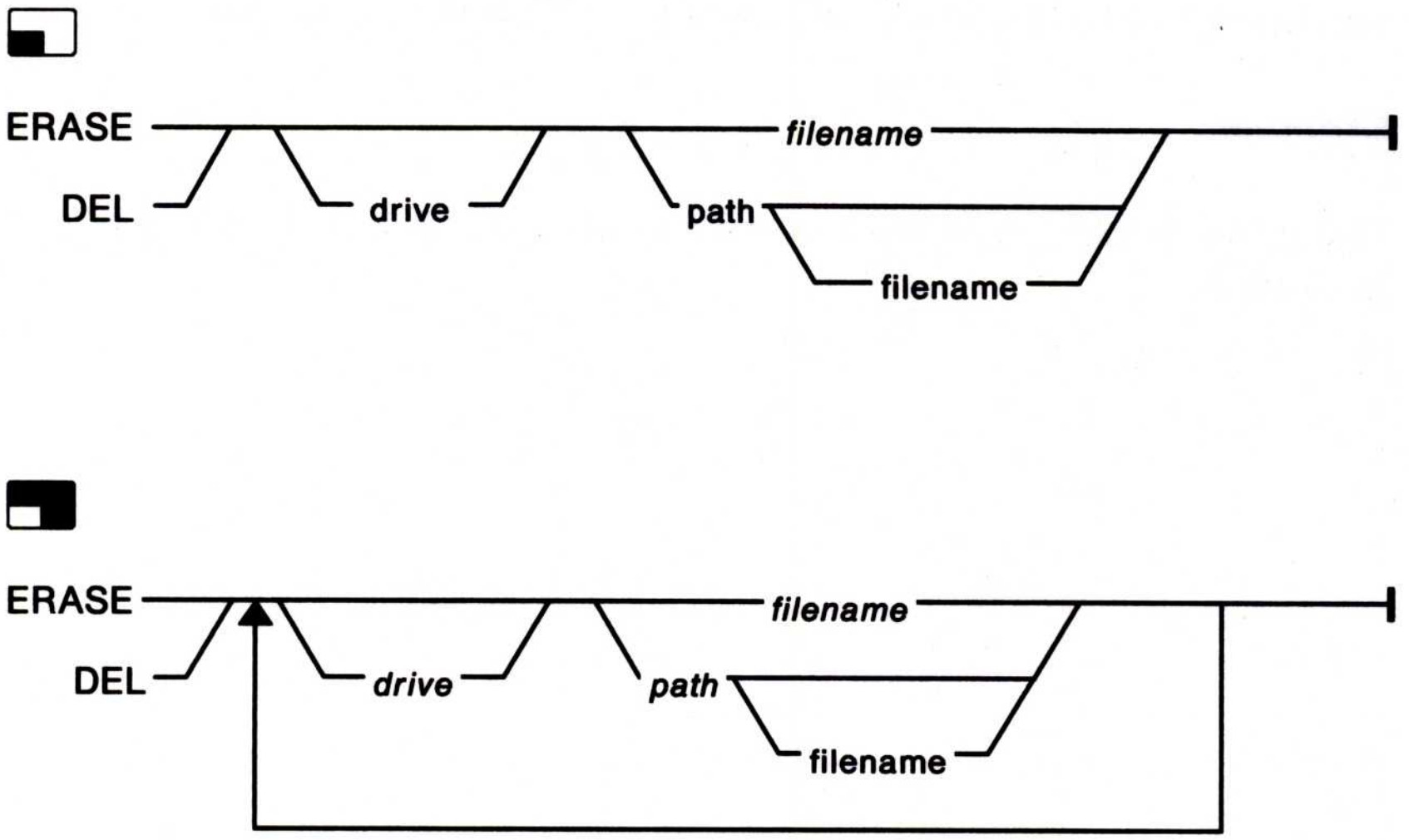
APPEND, PATH, SET

ERASE (delete files)

Purpose

Deletes one or more files.

Syntax



Remarks

In OS/2 mode, you can delete files on different drives and in different directories. For example, the command:

```
DEL C:*.TXT A:MEMO
```

deletes all the files with a .TXT extension in the current directory of drive C, and the file MEMO on drive A.

You can delete all the files in a directory with DEL. If you enter a path to a directory without a filename, you are requesting that all files in that directory be deleted. The prompt, **Are you sure (Y/N)?** appears. For example, if the root directory is your current directory, to delete all files in a directory named \MYDIR, enter:

```
DEL MYDIR
```

ERASE (delete files)

This is equivalent to entering:

```
DEL MYDIR\*.*
```

This does not remove the directory MYDIR nor delete any of its subdirectories.

Warning: The name of a file you want to erase should not be the same as that of a directory. The prompt **Are you sure (Y/N)?** alerts you that you are specifying all the files in a directory for deletion.

Example

The following example deletes the filename FILE.BAT from the diskette in drive A.

```
DEL A:FILE.BAT
```

EXIT (end a command processor)

Purpose

Ends the current command processor and returns to the previous one, if one exists.

Syntax



Remarks

This command can be used when you want to end the current command processor and return to the previously started command processor.

If you are running an OS/2 program, you must first end the program before typing EXIT. If no previous OS/2 command processor exists, typing EXIT returns you to the Program Selector.

If you are running a DOS program, EXIT returns you to the previous DOS command processor. If none exists, EXIT does nothing and you remain at the DOS command prompt. To return to the Program Selector, press Ctrl+Esc.

Related Commands

COMMAND, CMD

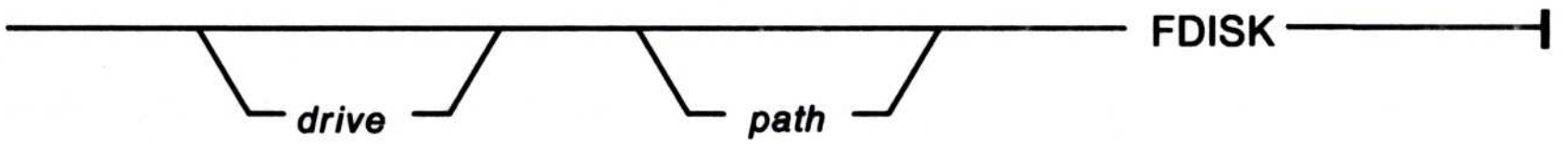
FDISK (manage partitions)

Purpose

Allows you to:

- Create a primary partition
- Create an extended partition
- Create a logical drive in the extended partition
- Change the active partition
- Delete a primary partition
- Delete an extended partition
- Delete a logical drive in the extended partition
- Display partition data
- Display logical drive information
- Select the next fixed disk drive for partitioning (for systems with more than one fixed disk).

Syntax



Remarks

FDISK is an external command with menus and screens that guide you through the tasks listed above. Highlighted here are the FDISK tasks:

- *Create partition* lets you specify the entire fixed disk be reserved for OS/2 (a maximum of 32Mb for a primary partition) or only part of it for OS/2 and the rest for one or more other operating systems.
- *Change active partition* allows you to specify which operating system to use upon system startup.
- *Delete partition* allows a primary or extended partition to be removed from the fixed disk. It should be noted that all data within the deleted partition will be lost.
- *Display partition data* lets you view information on how the fixed disk is partitioned.
- *Select next fixed-disk drive* allows multiple drives to be partitioned during one session.

FDISK (manage partitions)

No programs should be running when the following FDISK tasks are performed:

Creating a primary or extended partition
Deleting a primary or extended partition
Creating or deleting a logical drive

After you complete any of the above tasks, FDISK prompts you to press Ctrl+Alt+Del to restart the system.

Rules for Partitions and Logical Drives: A partition or logical drive must be greater than 32Kb.

A primary partition is required to start the system. A primary partition on a disk whose storage capacity exceeds 32Mb, must be located within the first 32Mb of storage.

Creating an extended partition with FDISK allows you to access media greater than 32Mb as multiple logical drives. Within this partition, you may create, delete, and display logical drives. A logical drive cannot exceed 32Mb because of the limitations of the File Allocation Table file system.

Consider the following requirements before creating extended partitions:

- An extended partition cannot be "started."
- An extended partition can be created on a drive that has a primary partition.
- An extended partition can be created on a drive that does not have a primary partition.
- An extended partition cannot be created on the startup drive unless a primary partition exists. An extended partition can be created on a drive that doesn't have a primary partition if it is not the startup drive.

FDISK (manage partitions)



Changing Partitions on the Startup Drive: If you want to change partitions on the startup drive once OS/2 has been installed, follow this procedure:

1. Insert Operating System/2 Diskette 1 into drive A and press Ctrl+Alt+Del. The system is started and the Program Selector menu is displayed.
2. Select the OS/2 command prompt from the Start a Program list.
3. Remove Diskette 1.
4. Insert the OS/2 diskette that contains FDISK and enter:

   ```
   FDISK
   ```

5. Follow the instructions that appear on your screen.

Note: No partition will be physically changed on the fixed disk until you press the Esc key while in the main menu of FDISK.

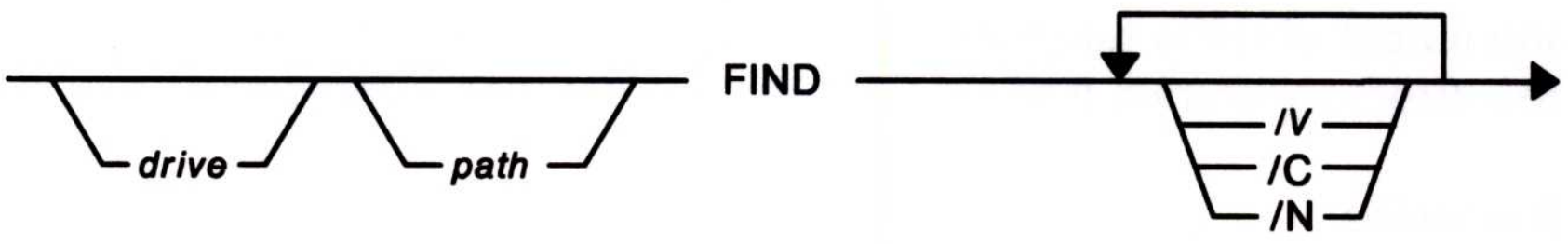
Warning: Any partitions you create with FDISK must be formatted before you can use them.

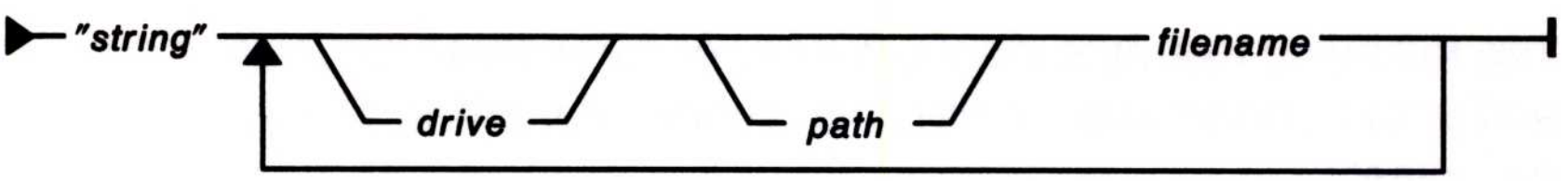
FIND (filter)

Purpose

Searches for a specific string of text in a file or files.

Syntax





Parameters

/V	Displays all lines *not* containing the specified string.
/C	Displays the count of lines that contain a match in each of the files. If /C is specified with /N, then FIND ignores /N. If /C is specified with /V, then FIND returns the count of lines *not* containing the string.
/N	Puts the relative line number in front of each matching line.
string	Specifies the string to search for. Remember to enclose the string in quotation marks.

Remarks

FIND is a external filter that takes as parameters a string (a group of characters) and a series of filenames. It displays all lines that contain a specified string from the files in the command line. If no filenames are typed, FIND takes input from its standard input (usually the keyboard), and displays all lines that contain the specified string.

The *string* must be enclosed in double quotes ("). Two single quotes in succession are equivalent to a double quote. An uppercase string does not match a lowercase string.

FIND (filter)

Global filename characters are not allowed in the filenames or extensions. The file is searched up to the first end-of-file character (Ctrl+Z)

With the piping feature (¦), you can cause FIND to receive its input from the output of another command. For example, if you enter:

```
DIR ¦ FIND "<DIR>"
```

the output of DIR is received as input by FIND, which then selects only the directory entries that contain <DIR> and displays them as output.

Examples

The following example displays all lines from BOOK1.TXT, BOOK2.TXT, and BOOK3 (in the same order) that contain the string "Fool's Paradise":

```
FIND "Fool's Paradise" BOOK1.TXT BOOK2.TXT BOOK3
```

The following example displays the names of all files in the current directory of drive C that *do not* contain the string "DAT":

```
DIR C: ¦ FIND /V "DAT"
```

For the following examples, assume the file PROG contains the following lines:

```
This is a
beautiful
day for a picnic.
```

The following example displays the lines in the file PROG that *do not* contain the string "f":

```
FIND /V "f" PROG
```

The result is:

```
---------- PROG
This is a
```

The following example displays the relative line number of the lines that contain the string "f":

```
FIND /N "f" PROG
```

The result is:

```
---------- PROG
[2]beautiful
[3]day for a picnic.
```

The following example displays a count of the number of lines in the file PROG that contain the string "a":

```
FIND /C "a" PROG
```

The result is:

```
---------- PROG: 3
```

The following example displays a count of the number of lines in the file PROG that do *not* contain the string "is":

```
FIND /V /C "is" PROG
```

The result is:

```
---------- PROG: 2
```

Related Commands

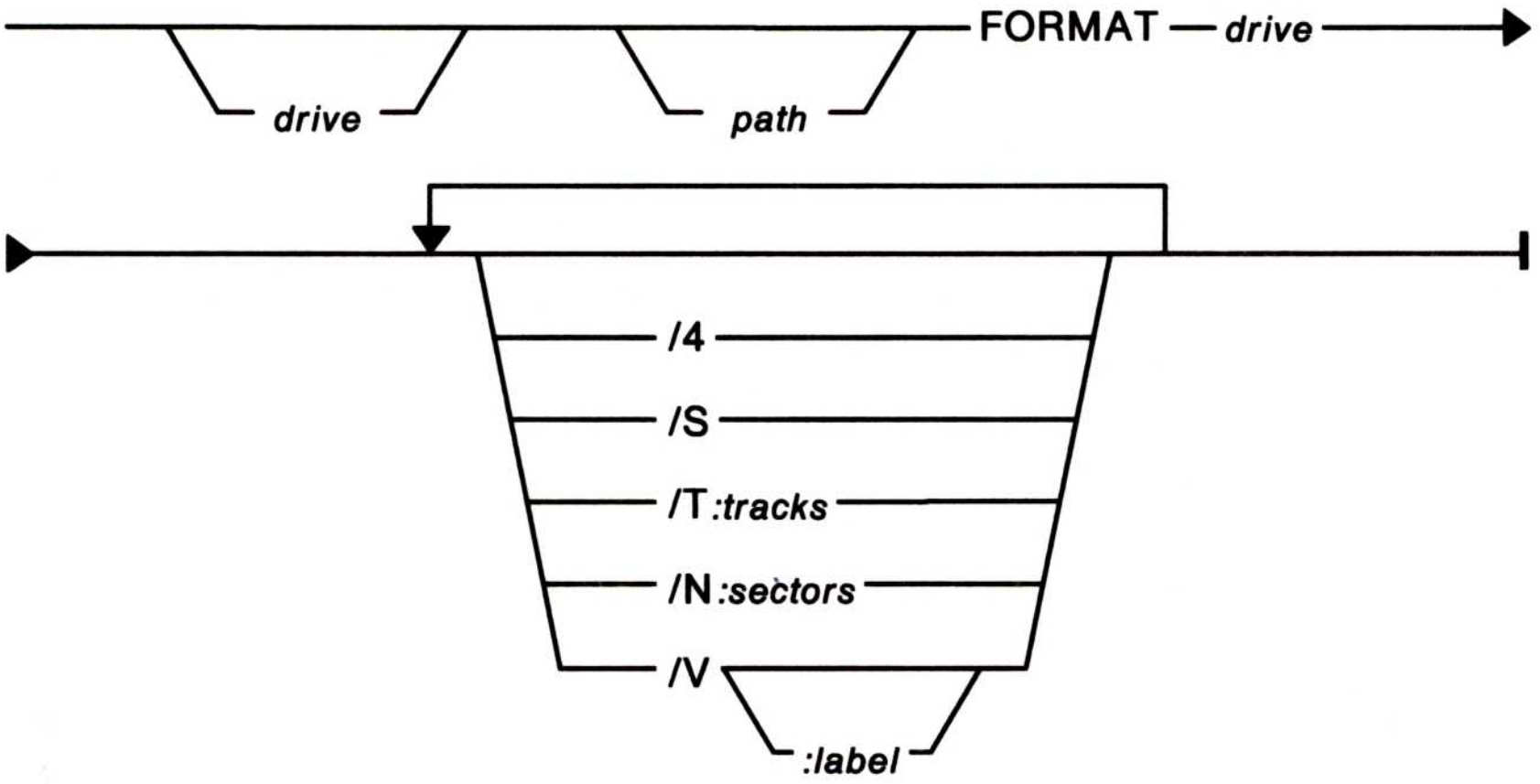
COMP

FORMAT (a fixed disk or diskette)

Purpose

Formats the disk in the specified drive to accept OS/2 files.

Syntax



Parameters

/4 Formats a 360Kb diskette in a 1.2Mb drive.

Warning: Diskettes formatted with this parameter for use on a 1.2Mb drive may not be reliably read or written in a 360Kb drive.

/S Copies the operating system files listed in the FORMATS.TBL file from the disk in the default drive to the newly formatted disk.

/T:*tracks* Formats the 3.5 inch diskette to the number of *tracks* specified. For 720Kb and 1.44Mb, this value is 80. If /T is not specified, the default is 80.

/N:*sectors* Formats the 3.5 inch diskette to the number of *sectors* per track specified. For 720Kb, this value is 9; for 1.44Mb, this value is 18. The default depends on the diskette drive being used. However, if you are using a

diskette that is less than the capacity of the diskette drive, you must specify a value for /N.

Note: Although the /T and /N parameters can also be used with 5.25 inch diskettes, their intended use is for 3.5 inch diskettes. See the discussion "Specifying Tracks and Sectors for Diskettes" on the next page.

/V:*label* Specifies the volume *label*. A volume label can be up to 11 characters long and is used to identify the disk. Blanks are not allowed in the volume label you specify. If you do not specify a volume label, you are prompted to do so after formatting is completed.

Remarks

This external command initializes the directory and file allocation tables on a disk. If no drive is specified, you will get a message telling you the drive letter must be specified.

CAUTION:
Be sure to set up an OS/2 partition on all fixed disks before formatting them. FORMAT does not recognize a fixed disk as being an OS/2 disk if an OS/2 or an IBM Personal Computer DOS partition does not exist. When FORMAT finds such a disk, it skips to the next disk and begins formatting that.

Copying the System Files: When you use the /S parameter, if the hidden system files do not reside in the root directory of the default drive, FORMAT prompts you to put a system diskette in:

- Drive A, if the default drive is nonremovable, or
- The default drive, if it is removable.

FORMAT notifies you of any other system files that are not transferred. Copy these files to the target disk after FORMAT has completed.

FORMAT (a fixed disk or diskette)

Warning: Since FORMAT expects all the system files to reside in the root directory of the default drive when you use the /S parameter, if any of them are not (for example, dynamic link libraries, whose location is specified with a LIBPATH statement in CONFIG.SYS), FORMAT is unable to find them. Thus, to format a disk and transfer the system when system files are not in the root directory, you must:

1. Format the disk without the /S parameter.
2. Use SYS to transfer the hidden files, IBMBIO.COM and IBMDOS.COM.
3. Use COPY to transfer the remaining system files.

Note: There are more OS/2 system files to copy than the standard three for IBM Personal Computer DOS. To see the names of system files that need to be copied, you can enter:

```
TYPE FORMATS.TBL
```

Specifying Tracks and Sectors for Diskettes: By default, the FORMAT command attempts to format the media to the maximum capacity of the diskette drive. The purpose of the /T and /N parameters allow you to format 3.5 inch diskettes to *less than* the maximum capacity of the diskette drive.

The following table lists the correct values for /T and /N, depending on diskette type:

Diskette Type	/N: Number of Sectors per Track	/T: Track Count	Root Directory Entries
5.25 360Kb	9	40	112
5.25 1.2Mb	15	80	224
3.50 720Kb	9	80	112
3.50 1.44Mb	18	80	224

For example, to format a 3.5 inch double-sided 720Kb diskette in a 1.44Mb drive, you must specify:

```
FORMAT A: /N:9
```

Note: Although you may use the /T and /N parameters for 5.25 inch diskettes, the /4 parameter is sufficient when formatting a 360Kb diskette in a 1.2Mb drive.

FORMAT (a fixed disk or diskette)

The following table shows the possible 3.5 inch drive type and diskette type combinations that can be used:

Drive Type	Diskette Type
1.44Mb	1.0Mb (720Kb formatted capacity) 2.0Mb (1.44Mb formatted capacity)
720Kb	1.0Mb (720Kb formatted capacity)

If you intend to transfer diskettes between computers that have diskette drives with different storage capacities, use only 1.0Mb diskettes formatted to 720Kb.

Warning: Do not attempt to format a diskette to an incorrect capacity. It will not store your data reliably. For example:

- Do not format a 3.5 inch 1.0Mb diskette to 1.44Mb using the FORMAT default in a 1.44 Mb drive.
- Do not format a 3.5 inch 2.0Mb diskette to 720Kb using /N:9.

Identifying IBM Diskettes: All 3.5 inch diskettes have a write-protect window (square cutout) in the lower left corner. A 1.0Mb diskette is usually labeled "1.0Mb capacity" or "2HC." A 2.0Mb diskette is usually labeled "2.0Mb capacity" and always has an additional square cutout in the lower right corner.

Note: In DOS mode, FORMAT ignores any drive reassignments made by ASSIGN, SUBST, and JOIN commands.

Example

You need to format a 360Kb diskette in a high capacity 1.2Mb drive. Enter the following:

```
FORMAT A: /4
```

Related Commands

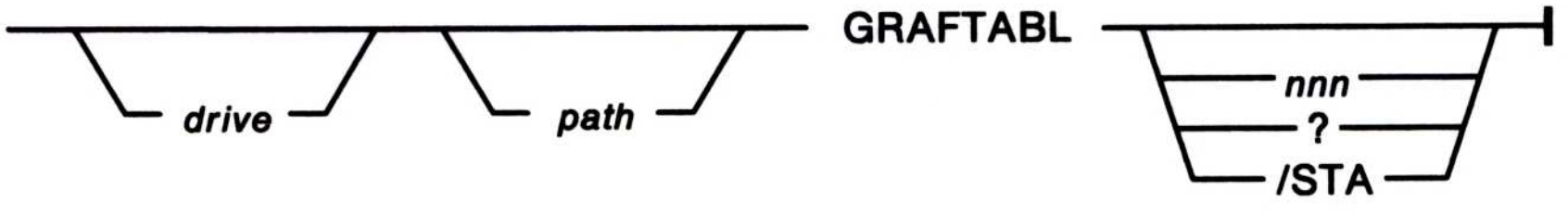
LABEL, SYS, VOL

GRAFTABL (graphics)

Purpose

Allows additional characters from a language code page to be displayed when using display adapters in graphics mode.

Syntax



Parameters

nnn A three-digit number, specifying the code page to be used, which can have a value of:

Value	Language
437	U.S. IBM PC
860	Portuguese
863	Canadian-French
865	Nordic

? Displays the number of the code page currently being used, and a list of code page options.

/STA Displays the number of the code page currently being used.

Remarks

GRAFTABL is an external command that allows the ASCII extended character set to be displayed when using display adapters for graphics. GRAFTABL only works for modes 4, 5, and 6 of the Color Graphics Adapter, Enhanced Graphics Adapter, and Video Graphics Array. Since GRAFTABL is a DOS command, it has no effect on OS/2 sessions.

Example

Suppose you enter the following command:

```
GRAFTABL 860
```

And then you enter:

```
GRAFTABL ?
```

The following message is displayed:

```
The Portuguese version (860) of the
graphic code page table is currently
loaded.
The acceptable GRAFTABL parameters are:

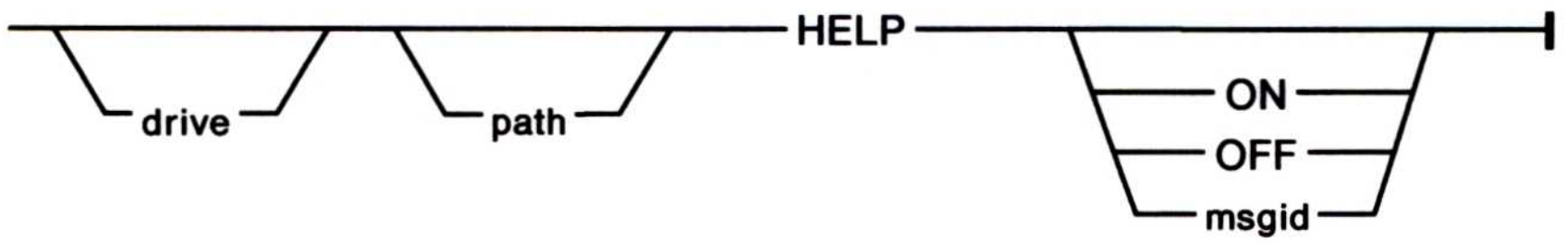
  /STA - Display the name and number of
       the current graphic code page
  ?    - Display this summary of
       parameters
  437  - U.S. IBM PC code page
  860  - Portuguese graphic code page
  863  - Canadian French graphic
       code page
  865  - Nordic graphic code page
```

HELP (help with messages)

Purpose

Provides a help line as part of the command prompt, a help screen, and information related to warning and error messages.

Syntax



Parameters

ON	Turns on the display of the help line
OFF	Turns off the display of the help line
msgid	Identifies the message for which additional information is being requested.

Remarks

HELP is an external command. If you enter HELP with no parameters at the command prompt, a help screen is displayed that tells you how to:

- Return to the Program Selector
- Switch to the next session
- Exit the current OS/2 session
- Get additional help on error and warning messages.

Similarly, if you enter:

```
HELP ON
```

a help line is displayed as part of your command prompt. It reminds you which keys to press to return to the Program Selector, switch to another session, and end a session. If you decide you no longer need this reminder, you can enter:

```
HELP OFF
```

HELP (help with messages)

Using HELP for Messages: When you receive a message from OS/2, it has a message identifier or *msgid*. For example, suppose in the process of entering a command, you misspell a filename, and no file exists with the name you have entered. You then receive a message from OS/2 that includes the misspelled filename, for example:

```
SYS1490: The system cannot find MYFIEL.TXT
```

The message identifier for this message is SYS1490. Each OS/2 message identifier consists of the letters SYS followed by a 4-digit message number. You do not have to type the letters SYS when requesting help for an OS/2 message, because SYS is the default. Nor is it necessary to type any leading zeroes. For example, suppose you receive the following error message:

```
SYS0003: The system cannot find the path specified.
```

To get help, you can type:

```
HELP 3
```

HELP then displays the original message followed by its associated help message. In this case, HELP would display:

```
SYS0003: The system cannot find the path specified.

EXPLANATION: The path named in the command does
not exist for the drive specified, or the path
was entered incorrectly.
ACTION: Retry the command using a correct path.
```

If the original message contains variable data, such as the name of a file, HELP substitutes three asterisks (***) for the variable data when it responds to your help request. An example of a message that would contain variable data is message 1490 shown on the previous page. Although the original message contains the filename you entered, when you ask for help by entering:

```
HELP 1490
```

HELP displays:

```
SYS1490: The system cannot find ***

EXPLANATION: The filename is incorrect or
does not exist.
ACTION: Check the filename and retry the
command.
```

HELP (help with messages)

HELP displays the following message if no help text is available for the *msgid* requested:

```
No help is available for the message
ID number requested.
```

If you type an incorrect *msgid*, you will get a message telling you the message number is incorrect.

Warning: If HELP is unable to find the help message file in the current directory, it searches the following directories for the file:

- Directories listed in the DPATH statement (OS/2 mode only)
- Directories listed in the APPEND statement (DOS mode only).

Note: The HELP command uses OSO001H.MSG to get help for system messages.

JOIN (join drives)

Purpose

Logically connects a drive to a directory on another drive.

Syntax

To join a drive to a directory on another drive:



To display joined drives currently in effect:



To delete joined drives currently in effect:



Parameters

drive drive *******directory*	The first drive is the one you want identified as **directory* on the second drive.
drive **/D**	Turns off a join in effect.

Remarks

You can use this external command to join a drive to a directory on another drive so that you can access a drive by a directory name instead of a drive letter. You can only join a drive at the level just below the root directory.

If the directory name does not exist, OS/2 creates a directory on the drive you specify. If the directory already exists, it must be empty for

the join to work. After you issue the JOIN command, the first drive name becomes incorrect, and if you try to use it, OS/2 displays an error message.

Examples

If you just type:

```
JOIN
```

OS/2 displays the current drives that are joined.

To join drive A to the path C:\DRIVEA, enter:

```
JOIN A: C:\DRIVEA
```

Drive A can now be accessed as the subdirectory \DRIVEA on drive C. To turn off this join, you would enter:

```
JOIN A: /D
```

Related Commands

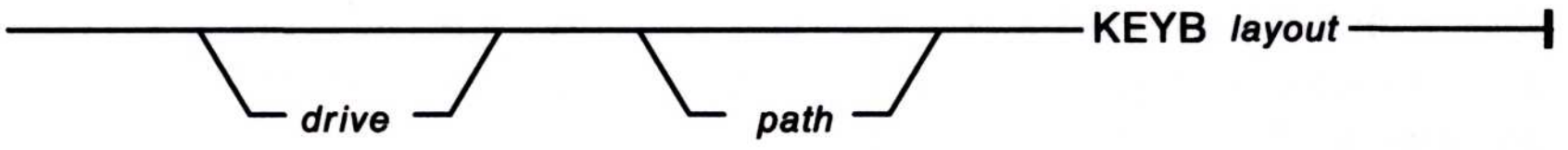
SUBST, ASSIGN

KEYB (keyboard layout)

Purpose

Selects a special keyboard layout to replace the default U.S. keyboard layout.

Syntax



Parameters

layout Specifies the keyboard layout. The values for *layout* are:

Layout	Keyboard
BE	Belgium
CF	Canada (French)
DK	Denmark
SU	Finland
FR	France
GR	Germany
IT	Italy
LA	Latin America
NL	Netherlands
NO	Norway
PO	Portugal
SP	Spain
SV	Sweden
SF	Switzerland (French)
SG	Switzerland (German)
UK	United Kingdom
US	United States

KEYB (keyboard layout)

Remarks

This is an external command that supports U.S. English and certain non-U.S. English keyboards.

To use the KEYB command, you must have a DEVINFO statement in your CONFIG.SYS file. A DEVINFO statement is required to identify any keyboard layout you want to specify with the KEYB command. If you specify a keyboard layout with KEYB that does not have a DEVINFO statement in your CONFIG.SYS file, you receive an error message. To see the syntax required for keyboards by the DEVINFO statement, refer to page 4-30.

Changing from one keyboard to another is global for all tasks in the system in both modes, but the command must be issued in OS/2 mode.

KEYB prepares the keyboard in the selected layout for the system code pages. A list of countries and their supported code pages can be found on page 4-11 under the discussion of the COUNTRY statement for the CONFIG.SYS file. If the code pages your system has been prepared for and the code pages for the country of the keyboard you select don't match, a message informs you of this. The message allows you to choose to have the default code page loaded (a "Y" response) or to end the KEYB command without loading a code page (an "N" response).

Warning: If you choose to have the default code page loaded, the keyboard may then be using a different code page from the one being used by the display; in which case, characters entered from the keyboard could be displayed incorrectly on the screen.

Note: Keyboard templates are provided, in alphabetic order by country, in Appendix D, "Keyboard Templates" on page D-1.

You can get accented letters by using nonescaping keys, that is, keys that allow characters (such as accents) to be typed without changing the print position. For a list of the allowable combinations of accent keys and letter keys for different countries, see Appendix E, "Accented Characters" on page E-1. To put an accent over a letter, press the accent character (the nonescaping key), and then press the appropriate letter key. To use the accent character by itself, press the accent key, and then press the space bar. With the

Canadian French keyboard you can also get the accent character by pressing the accent key twice.

In most cases, if the nonescaping key plus letter key combination is not allowed, the accent followed by the character key you pressed is displayed, and you hear a beep. However, with the Canadian French keyboard, you only hear a beep and nothing is displayed.

Example

Suppose you have the following statement in your CONFIG.SYS file:

```
DEVINFO=FR KEYBOARD.DCP
```

Then to switch to the French keyboard, you can enter this command at the OS/2 command prompt:

```
KEYB FR
```

Related Commands

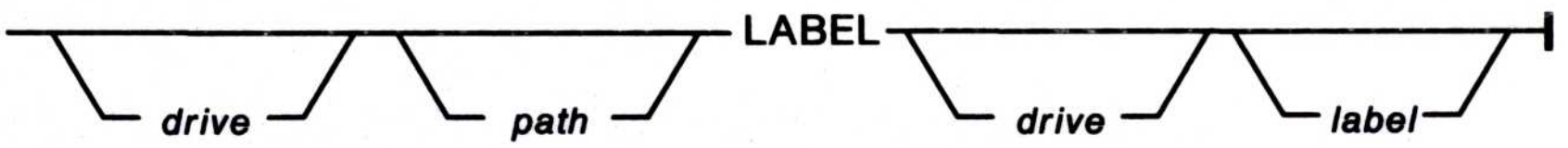
COUNTRY statement in CONFIG.SYS, DATE, TIME

LABEL (volume label)

Purpose

Creates or changes the volume identification label on a disk.

Syntax



Parameters

label Specifies the volume label that identifies the disk. This can be up to 11 characters. All characters acceptable in filenames are acceptable in the volume label. Unlike filenames, however, the volume label does not contain a period between the eighth and ninth characters.

Remarks

LABEL is an external command you may use to name your disk or diskettes. If you do not specify a volume label, LABEL prompts:

```
 The volume label in drive X is XXXXXXXXXXX.
Enter a volume label of up to 11 characters
or press Enter for no volume label update.
```

If the disk does not already have a volume label, LABEL prompts you for one. Type the volume label that you want and press the Enter key. If you press Enter without entering a label the volume label remains unchanged.

In DOS mode, LABEL should not be used on drives specified in SUBST ASSIGN, or JOIN commands.

Examples

In the following example, a volume label called ADDRESS is created on the diskette in drive A:

```
LABEL A:ADDRESS
```

In this example, fixed disk C is given the volume label FIXEDISKC.

```
LABEL C:FIXEDISKC
```

In this example, the volume label of the diskette in drive A is changed from ADDRESS to PROGRAMS.

First you enter:

```
LABEL A:
```

Then you are prompted:

```
 The volume label in drive A is ADDRESS.
Enter a volume label of up to 11 characters
or press Enter for no volume label update.
```

Enter:

```
PROGRAMS
```

Related Commands

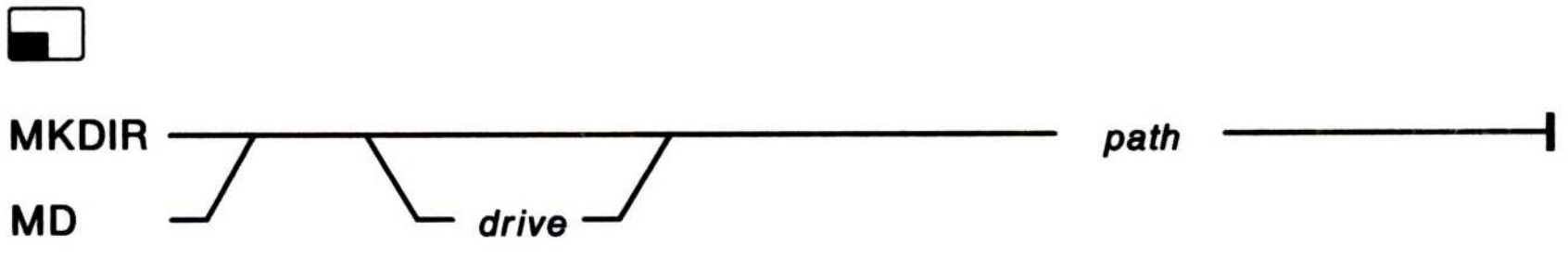
FORMAT, VOL

MKDIR (make directory)

Purpose

Makes a new directory.

Syntax





Remarks

This command creates a multilevel directory structure, which is helpful in keeping related program or data files together.

Example

The command:

```
MKDIR \GUIDE
```

creates a \GUIDE subdirectory directly under your root directory. To create a directory named ART beneath the \GUIDE directory, type:

```
MKDIR \GUIDE\ART
```

When you create directories with MKDIR, they always appear under your current directory unless you specify a different path with MKDIR. Suppose in the above example you were in the \GUIDE directory. To create the directory \GUIDE\ART\, you can just enter:

```
MKDIR ART
```

MKDIR (make directory)

In OS/2 mode, you can create more than one subdirectory at a time by specifying multiple paths. For example, to create the directories \GUIDE and \GUIDE\ART with one command line, type:

```
MKDIR \GUIDE \GUIDE\ART
```

Note that drive parameters may be included when specifying multiple paths.

Related Commands

CHDIR, RMDIR, DIR, TREE

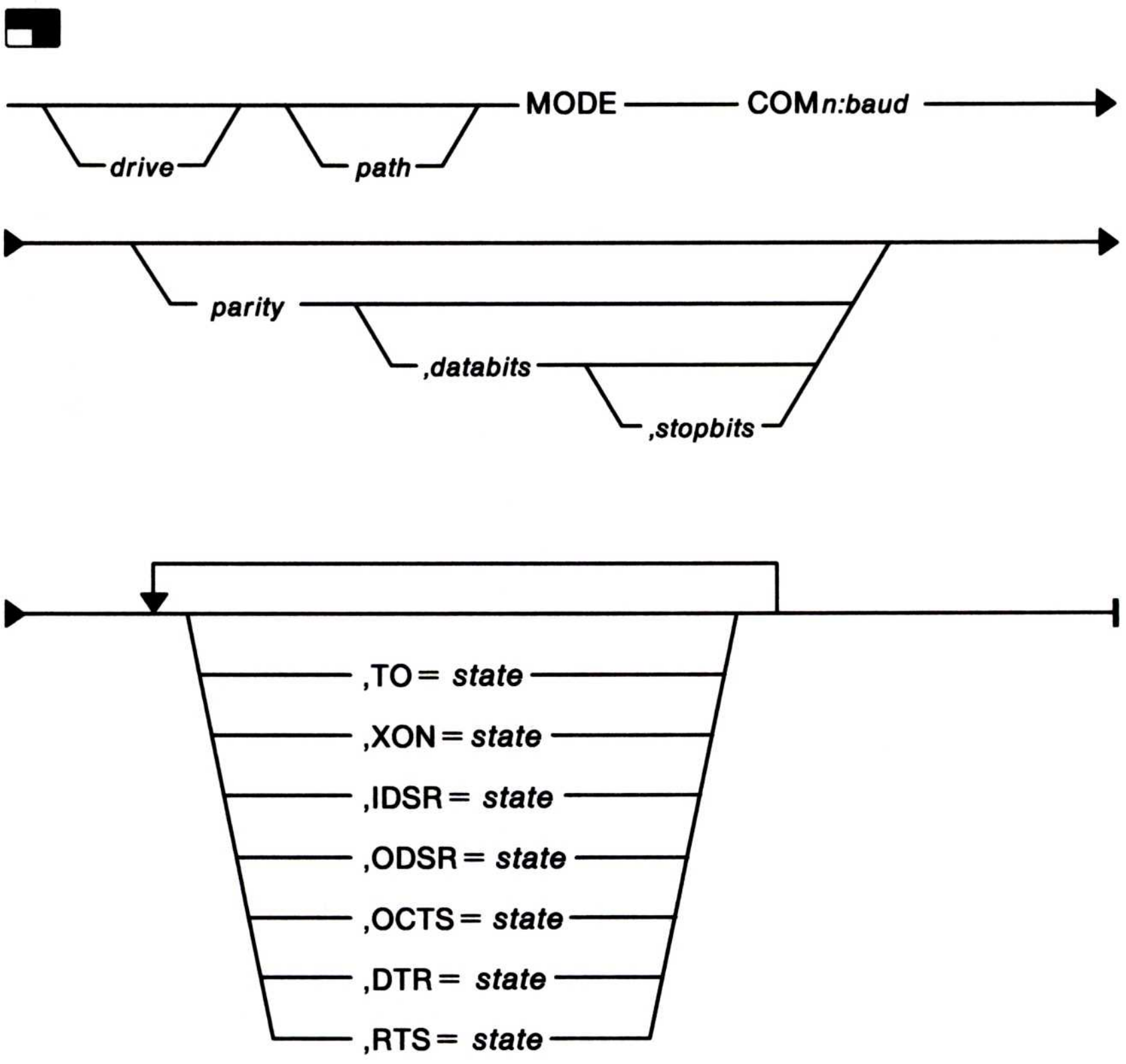
MODE (device control)

Purpose

Sets operation modes for devices.

Syntax

To set OS/2 asynchronous communications modes:

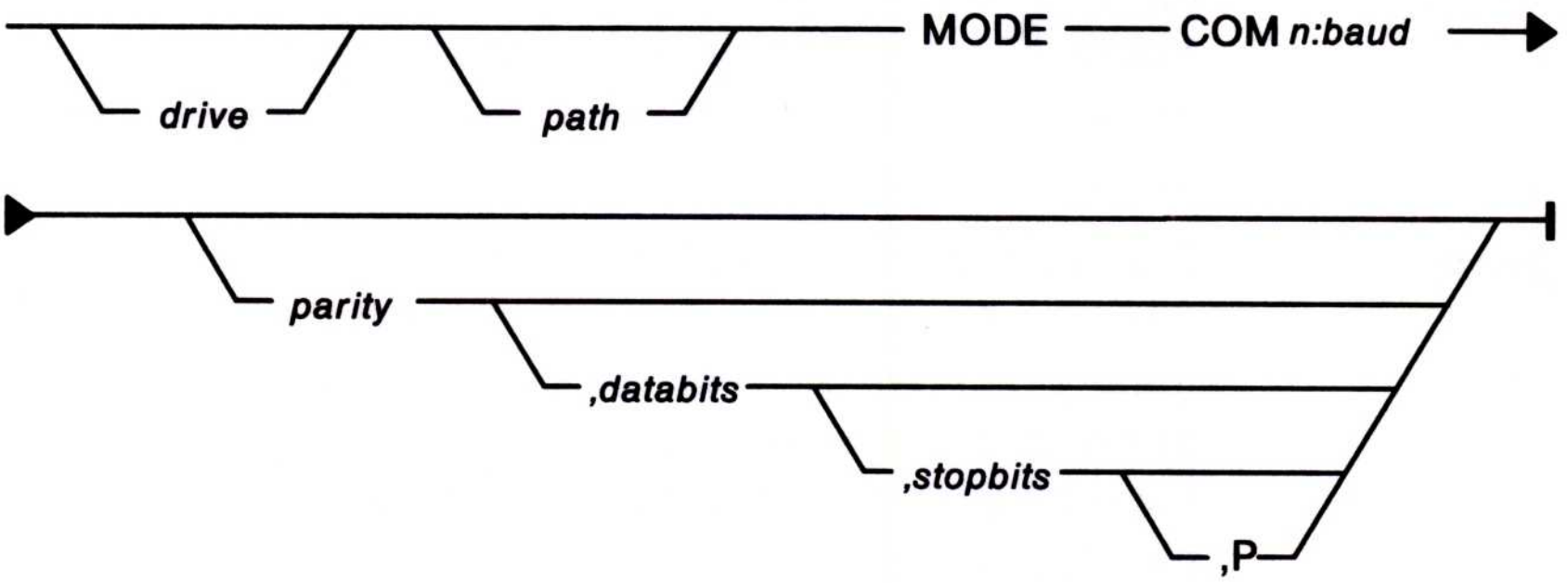


Note: The use of MODE to set COM port modes is allowed only when the Base Asynchronous Communications Device Driver (COM0*x*.SYS) is installed. For more information on this device driver, see "COM0x.SYS Device Driver" on page 4-17. Also refer to the DEVICE statement on page 4-13.

MODE (device control)

Syntax

To set DOS asynchronous communications modes:



Note: The *parity, databits, stopbits*, and P parameters are positional. When a positional parameter is omitted, but the one that follows it is specified, its position must be designated by a comma. For example, to accept the defaults for *parity, databits,* and *stopbits* and specify the P parameter, you can enter:

```
MODE COM1:2400,,,P
```

To query asynchronous communications port status in OS/2 mode:



Setting asynchronous communications modes: To set asynchronous communications modes, you may issue the command in either DOS or OS/2 mode. However, some serial devices require additional parameters that must be set in OS/2 mode. Therefore, it is recommended that all asynchronous communications settings be done in OS/2 mode. While in OS/2 mode, you can display the current asynchronous communications port settings by querying, for example:

```
MODE COM1
```

Output of a MODE COM*n* query can be redirected, modified, and used as input to another MODE command.

MODE (device control)

Parameters

COM*n:baud* — *n* is asynchronous communications port numbers 1 through 8; *baud* is the transmission rate, which may be: 110, 150, 300, 600, 1200, 2400, 4800, 9600, or 19200. (Only the first two characters are required except for 19200, which requires the first three.) The default is 1200.

parity — Can be: N (none), O (odd), E (even), M (mark), or S (space). None means there is no parity bit. Odd means odd parity. Even means even parity. Mark parity means the parity bit is always 1. Space parity means the parity bit is always 0. If parity is not specified, the default is E.

databits — 5, 6, 7, or 8 bits of data, with the default of 7.

stopbits — 1, 1.5, or 2. Either 1 or 1.5 is valid if *databits* is specified as 5. If *stopbits* is specified as 1.5, then only 5 is valid for *databits*. If *stopbits* is omitted, the default is 2 when *baud* is 110; otherwise the default is 1.



P — This parameter specifies a time out value of approximately 30 seconds in DOS mode for IBM Personal Computer DOS programs that go directly to the hardware. These programs need to know the port address, and so you must run the SETCOM40 command first before printing or plotting. The P parameter does not provide infinite retry as in previous versions of IBM Personal Computer DOS. The TO parameter described below provides similar function for OS/2 mode.



TO=*state* — ON enables Write Infinite Timeout processing. If OFF is specified, then normal time out processing is done. If this parameter is not specified, the state of TO is OFF. MODE will ignore it.

XON = *state* — ON enables and OFF disables automatic transmit flow control. If this parameter is not specified, the state of XON is OFF.

IDSR = *state* — ON enables and OFF disables input handshaking using Data Set Ready (DSR). If this parameter is not specified, the state of IDSR is ON.

ODSR = *state* — ON enables and OFF disables output handshaking using Data Set Ready (DSR). If this parameter is not specified, the state of ODSR is ON.

OCTS = *state* — ON enables and OFF disables output handshaking using Clear To Send (CTS). If this parameter is not specified, the state of OCTS is ON.

DTR = *state* — ON enables Data Terminal Ready (DTR), OFF disables DTR, and HS enables DTR handshaking. If this parameter is not specified, the state of DTR is ON.

RTS = *state* — ON enables Ready To Send (RTS), OFF disables RTS, HS enables RTS handshaking, and TOG enables RTS toggling. If this parameter is not specified, the state of RTS is ON.

The following asynchronous communications device driver parameters are reset by MODE every time it is setting COM parameters when processing in OS/2 mode. When processing in DOS mode, the states of these parameters remain unchanged:

- Output handshaking using DCD (data carrier detect) is disabled.
- Automatic receive flow control (XON or XOFF) is disabled.

Redirection of parallel printer output to a serial device is not supported by the MODE command. However, this function is supported by the SPOOL command. See page 2-126.

MODE (device control)

Example

Suppose you have a serial printer attached to the COM2 port. You want to make sure that it interfaces with the port before printing. To query the status of the asynchronous communications port, enter the following at the OS/2 prompt:

```
MODE COM2
```

If you know that your printer has been set to the defaults,

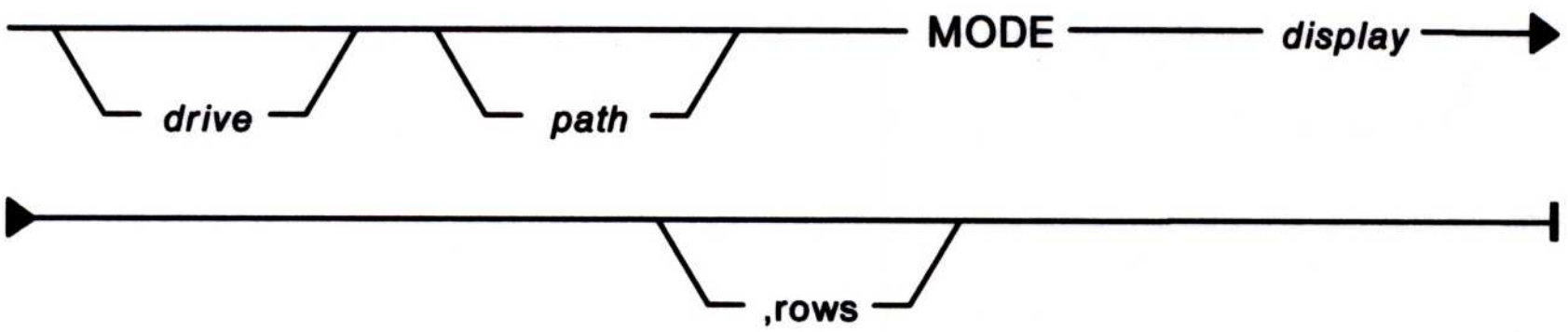
```
COM2:12,E,7,1
```

then there is no need to issue the MODE command before printing.

MODE (device control)

Syntax

To set display modes:



Setting display modes

Parameters

display One of: 40, 80, BW40, BW80, CO40, CO80, or MONO. For each of these, 40 specifies a display width of 40 characters per line and 80 specifies a display width of 80 characters per line. BW and CO refer to a color graphics monitor adapter with color disabled (BW) or enabled (CO). MONO specifies the monochrome mode, which always has a display of 80 characters per line.

rows Number of rows to select. Valid rows are 25, 43, or 50, depending on the display adapter attached. For example, with an enhanced display adapter, you may specify a 43-line display:

```
MODE CO80,43
```

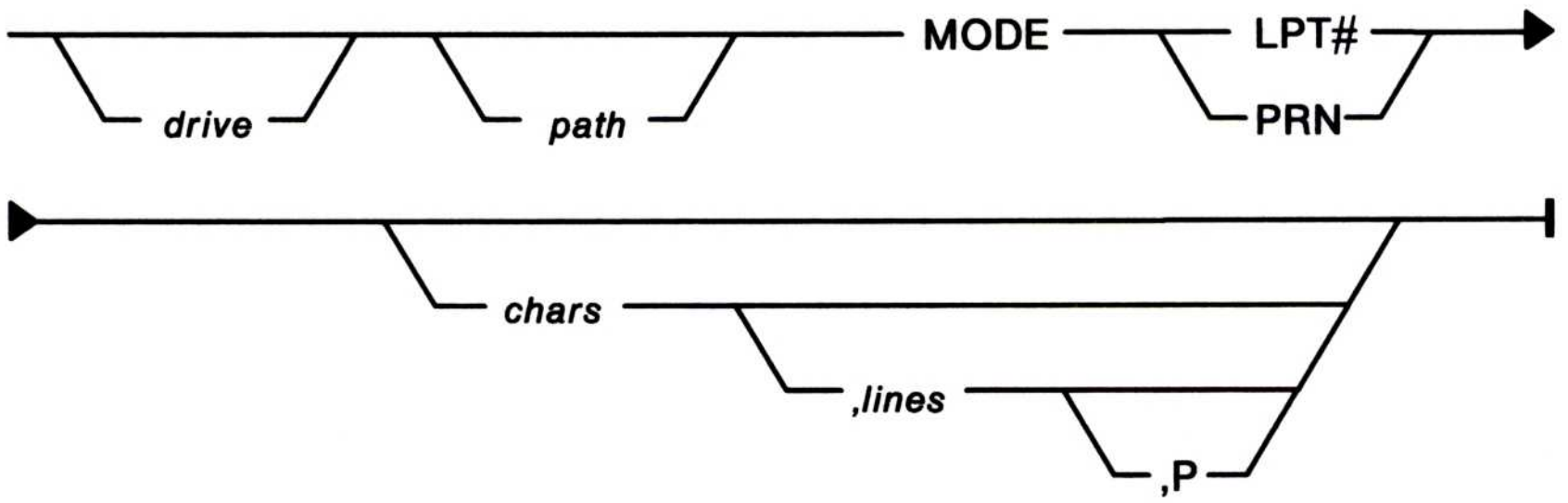
If *row* is not specified, the default is 25.

Note: The display mode that is set with the MODE command affects only the current session in which MODE is issued. This includes any programs or command processors running in this session.

MODE (device control)

Syntax

To set parallel printer modes:



Note: The *chars*, *lines*, and P parameters are positional. When a positional parameter is omitted, but the one that follows it is specified, its position must be designated by a comma. See the first example under "Examples."

Parameters

LPT#	# is printer number 1, 2 or 3. PRN is an acceptable device name for LPT1
chars	Either 80 or 132 characters per line. The default is 80.
lines	Either 6 or 8 lines per inch vertical spacing. The default is 6.
P	Continuously try to send output to the printer if a time out error occurs.

Note: Once you have issued the MODE command and specified values for either *chars* or *lines* or both these parameters, their defaults are the last valid values you entered for them.

Examples

If you want to accept the system defaults for characters and lines and specify the P parameter the first time you issue the MODE command for your printer, remember to specify commas for the omitted parameters. For example, you can enter:

```
MODE LPT1 ,,P
```

Suppose you already have set up your printer for 132 characters per line and eight lines per inch for vertical spacing, and have specified P to have the system continuously try to send output to the printer if a time out occurs. The command you entered looked this:

```
MODE LPT2 132,8,P
```

To change only the number of lines per inch for vertical spacing to 6, you can enter:

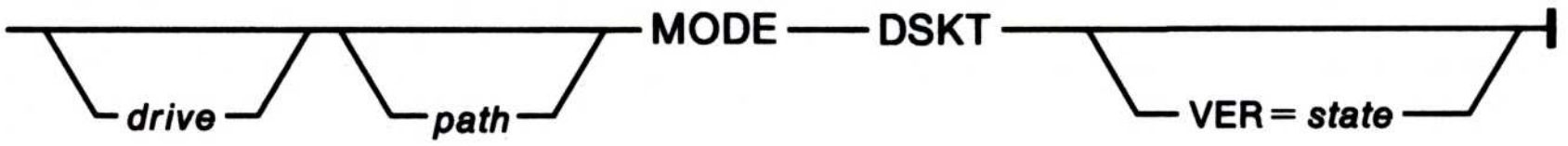
```
MODE LPT2 ,6,P
```

Your printer will still be set to 132 characters per line.

MODE (device control)

Syntax

To set diskette I/O write verification or query its status:



Parameters

state VER=ON turns on diskette verification; VER=OFF turns off diskette verification.

Setting Diskette Verification: When you issue:

```
MODE DSKT VER=ON
```

verification is done for all diskette I/O writes for all sessions. This is different from the function offered by the VERIFY internal command. See the VERIFY command on page 2-147.

Note: If you do not have a fixed disk, a diskette (any diskette) must be present in diskette drive A for the MODE command to turn on or off diskette verification.

Example

To query the status of diskette verification, enter:

```
MODE DSKT
```

To turn off diskette verification, enter:

```
MODE DSKT VER=OFF
```

Related Commands

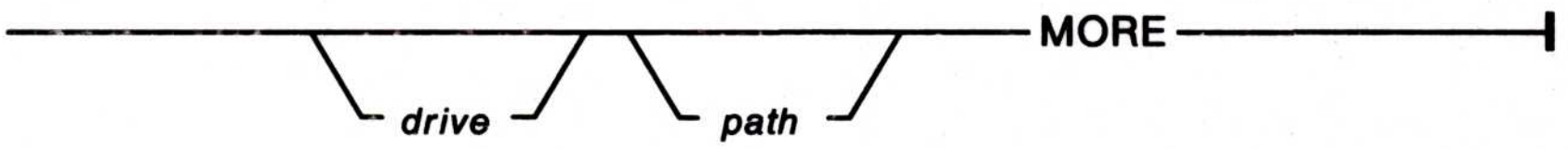
VERIFY

MORE (filter)

Purpose

Sends output to the standard output device (usually the display) one full screen at a time.

Syntax



Remarks

This external command is useful for viewing long files. MORE reads from standard input, such as the output from a command entered from your keyboard, and displays one screen of information at a time. The MORE command then pauses and displays the --**More**-- message at the bottom of your screen.

Pressing the Enter key displays another screen of information. This process continues until all the data has been read.

Examples

With the piping symbol (¦), you can cause MORE to receive its input from the output of another command. For example, suppose you enter:

```
TYPE MYFILES.NEW ¦ MORE
```

The output of TYPE MYFILES.NEW, instead of going directly to the screen, is received as input by MORE, which then displays the contents of MYFILES.NEW, one screenful at a time.

You can achieve the same result as above by using the redirection symbol (<) to redirect input. For example, if you enter:

```
MORE <MYFILES.NEW
```

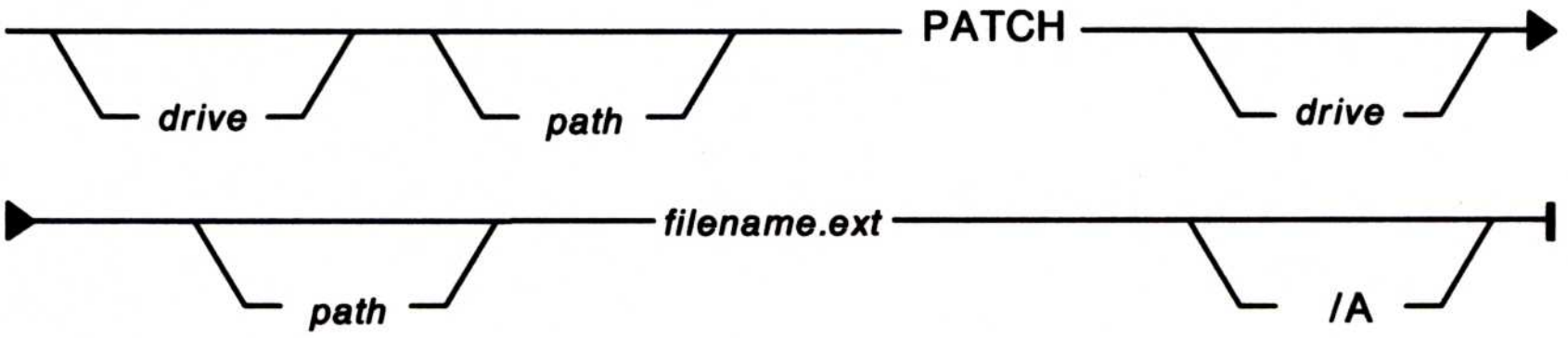
MORE receives its input from MYFILES.NEW.

PATCH

Purpose

Allows you to apply IBM-supplied patches to make fixes to code.

Syntax



Parameters

/A Specifies automatic mode. In this mode the filename is a file containing instructions for patching one or more files.

Remarks

PATCH is an external command that has prompts that guide you through the tasks. Interactive operation is the default.

If you select the option to automatically apply patches shipped by IBM to make fixes to IBM-supplied code, verification will be performed before the patch is applied. Verification may not be done on non-IBM-supplied patches. PATCH should be used by those who understand the need for a patch, how to make a patch, and the effect the patch will have on the operation of a program.

If you select the option to manually apply a patch, you will be requested to supply an offset where the patch is to be made. PATCH will display the contents of the location specified by the offset and allow you to enter the patch. Both the offset and the patch contents are expected to be entered in hexadecimal.

When you supply a hexadecimal offset, the 16 bytes at that offset will be displayed and you can change from one up to all 16 bytes or you

can quit without changing any. The cursor is initially positioned on the first byte. You can type one or two hexadecimal digits to change this byte or press the spacebar to leave the byte unchanged and move to the next byte. The Backspace key can be pressed to move the cursor to the previous byte if a mistake was made. You can continue changing bytes until Enter is pressed. If the cursor is moved past the 16th byte, the next 16 bytes are displayed, allowing these bytes to also be changed.

After Enter is pressed, the patch information is saved in storage. Then the following prompt is displayed:

```
Do you want to continue patching filename (Y/N)?
```

If the response is **Y**, PATCH prompts for an offset. Any more patch requests are also saved in storage. When all patches have been entered, they are displayed on the screen and you are asked if they should be applied. If you respond **Y**, all of the saved patch requests are written to the disk in the same order they were entered.

If a patch is necessary, automatic mode (/A) allows PATCH to get the information needed from a patch information file received from IBM.

Example

The following example uses the automatic mode and information contained in MYPATCH:

```
PATCH MYPATCH /A
```

Following are the prompts and messages that will appear on the screen in interactive mode.

```
Patching filename


End of file is at XXXX
Enter the hexadecimal offset of patch: _
```

This is the prompt for the offset of a patch. If Esc is pressed or Enter is pressed without an offset, a **Do you want to continue patching filename (Y/N)?** prompt will be displayed.

```
XXXXXXXX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX
```

This is displayed when prompting for a patch. The first field is the offset followed by the bytes at that offset. The cursor is initially positioned on the first byte. You can enter hexadecimal digits to change

that byte or press the spacebar to leave it unchanged and move the cursor to the next byte. If the cursor is moved past the 16th byte, the next 16 bytes in the file are displayed and may be changed in the same manner. This process repeats until Enter or Esc is pressed. If Esc is pressed, any changes made on the current line are ignored. When Enter or Esc is pressed, a prompt is displayed:

```
Do you want to continue patching filename (Y/N)? _
```

If the response to this is **Y**, the prompt for an offset is displayed. If the response is **N**, the entered patches are displayed. If no patches were entered, PATCH returns to the command prompt.

```
Patches entered for filename

XXXXXXX   XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX
XXXXXXX   XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX
XXXXXXX   XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX

Do you want these patches applied to filename (Y/N)? _
```

This is the display of entered patches and the prompt for confirmation. If the response is **Y**, the patches are written to the file in the same order as they are displayed, and the following message appears:

```
Patches applied to filename
```

If the response is **N**, no patches are written, and the following message is displayed:

```
No patches applied.
```

PATCH returns to the command prompt.

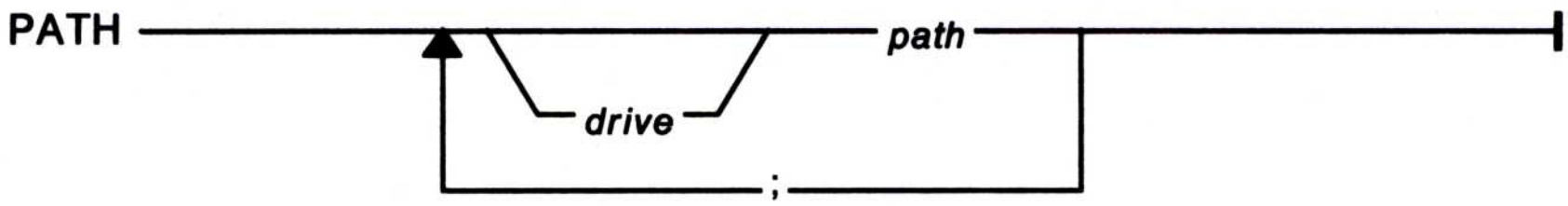
PATH (search path)

Purpose

Sets a search path for commands.

Syntax

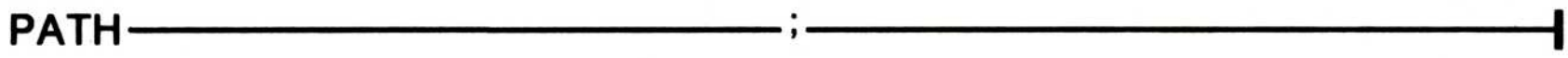
To specify paths to commands and programs:



To display paths currently in effect:



To delete usage of the PATH command:



Remarks

PATH is a system environment variable. If the system cannot find an external command or program in your current directory, it queries the environment for a value for PATH. Application programs can also query the environment for PATH's value and, depending on what they find, change their behavior. For more information on the OS/2 and DOS mode environments, see the SET command on page 2-117.

If you enter the PATH command without specifying any options, the current path is displayed, if one exists.

You can tell the system to search more than one path by specifying several paths separated by semicolons. OS/2 searches the paths in the order specified in the PATH command.

PATH (search path)

System installation creates a batch file called OS2INIT.CMD, which runs each time you select the OS/2 command prompt from the Program Selector. It contains this PATH command:

```
PATH C:\;C:\OS2;C:\OS2\INSTALL;
```

System installion also creates a batch file called AUTOEXEC.BAT, which runs the first time you select the DOS command prompt from the Program Selector. It contains this command:

```
PATH C:\;C:\OS2;
```

For more information about these special batch files, see "Special Batch Files" on page 3-4.

Examples

If you type:

```
PATH ;
```

OS/2 resets the search path to null. This means that OS/2 only searches the current directory for external commands.

In OS/2 mode, if you enter the following at the command prompt:

```
PATH %PATH%A:\PGM
```

OS/2 adds A:\PGM to the existing path known to the current command processor. In this example, the existing path to program files is entered as the replaceable parameter %PATH%, and it is assumed that your current path ends with a semicolon. For more information on using replaceable parameters at the OS/2 command prompt, see the SET command on page 2-117.

Related Commands

APPEND, DPATH, SET

PRINT

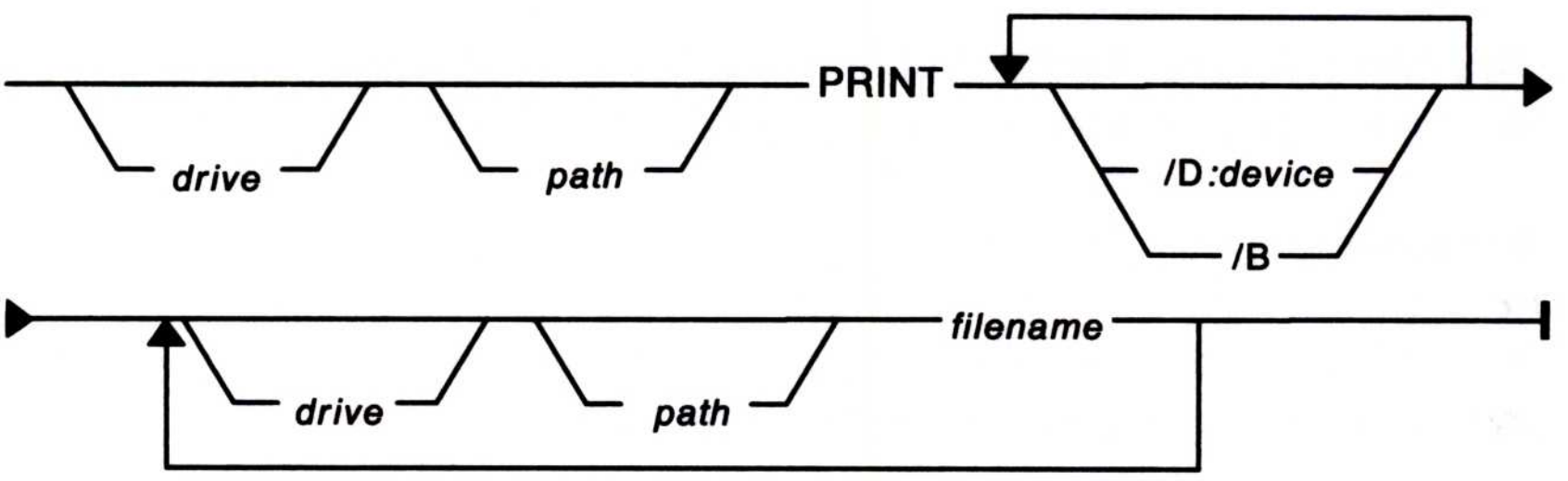
Purpose

Prints or cancels printing of one or more files.

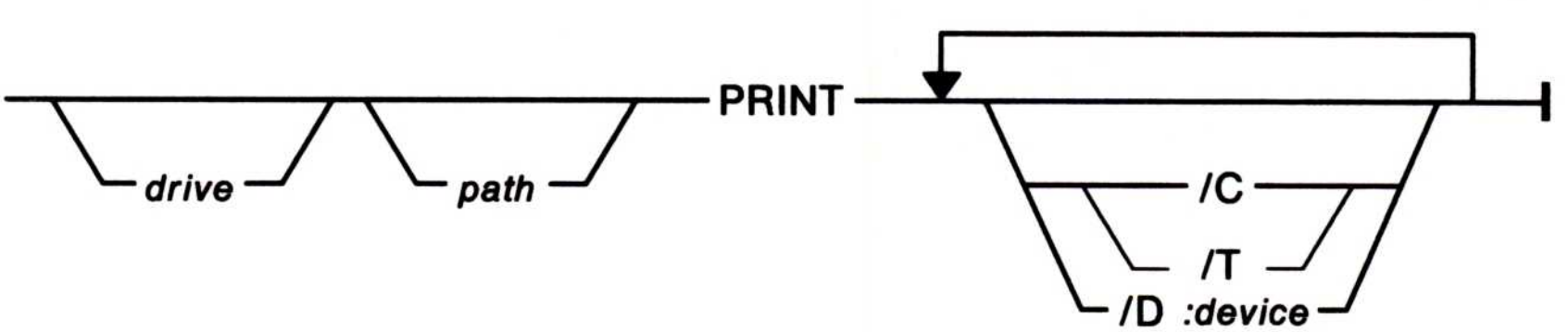
Note: Before using the PRINT command you should start the print spooler. This ensures that data going to the printer from concurrently running application programs is not intermixed when it is printed. See the SPOOL command on page 2-126.

Syntax

To print one or more files on a specified device:



To cancel the printing of the current file or all files waiting to be printed on a specified device:



Parameters

/D:device Specifies the print device. If not specified, the default device is LPT1. The acceptable device names are PRN, LPT1, LPT2, and LPT3.

PRINT

/B Causes Ctrl+Z characters encountered not to be interpreted as end-of-file indicators. The entire file is printed as is, without end-of-file character processing.

/C Cancels the currently printing file if the SPOOL command is active for the specified device.

/T Cancels all files waiting to be printed if the SPOOL command is active for the specified device.

Remarks

Neither the /C or /T parameter can be specified with the /B parameter. A filename cannot be specified with the /C or /T parameter. The /C and /T parameters cannot be used together.

You can only use the PRINT command if you have a printer attached to your computer. If spooling is required, use the SPOOL command.

Example

Assuming the print spooler is active and you wish to cancel the files in the queue for a specified printer, you can enter:

```
PRINT /D:LPT1 /T
```

Related Commands

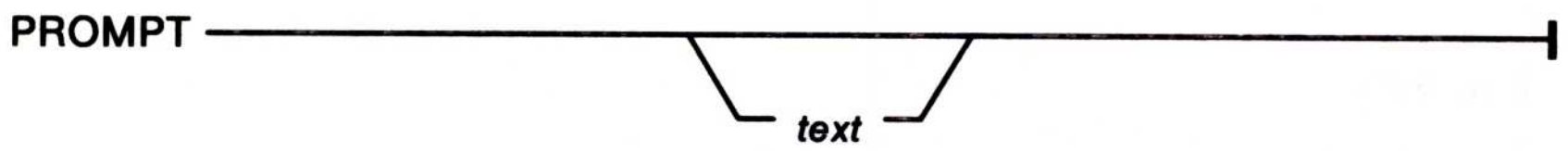
SPOOL

PROMPT (command prompt)

Purpose

Changes the command prompt.

Syntax



Parameters

text May contain any of the following characters to specify special prompts. You must precede each of the characters with a dollar sign ($):

$	The '$' character
_	A carriage return-line feed sequence
A	The '&' character
B	The '|' character
C	The '(' character
D	The current date
E	ASCII code 27 (escape)
F	The ')' character
G	The '>' character
H	The backspace symbol
I	The help line
L	The '<' character
N	The default drive
P	The current directory of the default drive
Q	The '=' character
S	A space (leading only)
T	The current time
V	The version number

PROMPT (command prompt)

Remarks

This command allows you to change the default system prompts. The default DOS system prompt is the default drive letter and current directory followed by the ">" symbol. For example, if you are in the root directory of the default drive, your DOS command prompt looks like this:

```
C:\>
```

The default OS/2 system prompt is the current directory of the default drive enclosed within bracket symbols "[]." For example, if you are in the \OS2 directory of the default drive, your OS/2 command prompt looks like this:

```
[C:\OS2]
```

Note: Although the prompt may be established with the SET command, the syntax used with SET is different from the syntax used with PROMPT. See page 2-117 for an explanation of these differences.

Example

If you have enabled extended keyboard and display support, you can use escape sequences in your prompts. For example:

```
PROMPT      $e[7m$n:$e[0m
```

sets the prompts in reverse video mode and returns to normal video mode for other text.

Note: Extended keyboard and display support is enabled differently for OS/2 and DOS modes. To enable support in DOS mode, refer to the DEVICE=ANSI.SYS statement for CONFIG.SYS on page 4-16. For information on support in OS/2 mode, see the ANSI command on page 2-7.

Related Commands

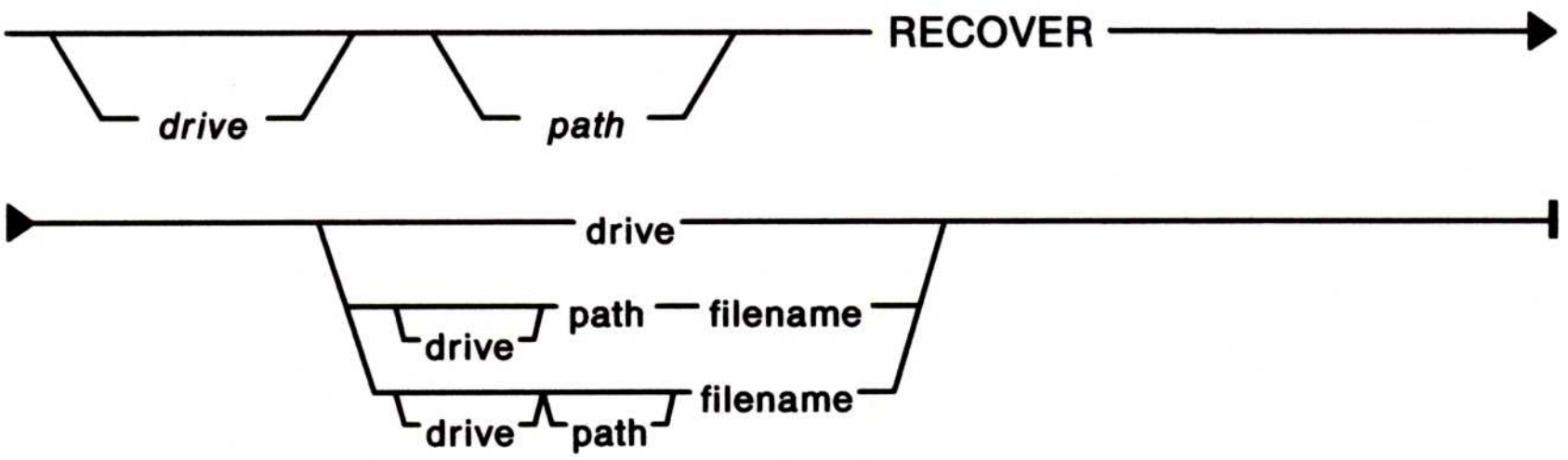
DATE, TIME, VER

RECOVER (recover files)

Purpose

Recovers files from a disk containing defective sectors.

Syntax



Remarks

RECOVER is an external command used to recover a file that contains a bad sector (minus the data in the bad sector). Or, all the files on the disk can be recovered if the directory has been damaged.

DOS Mode Only: This command cannot be used on drives involved with the SUBST or JOIN command.

Examples

To recover a particular file, enter its name. If necessary, include its drive and path. For example, to recover MYFILE.TXT in the root directory of your A drive, you can enter:

```
RECOVER A:\MYFILE.TXT
```

This will cause OS/2 to read MYFILE.TXT sector by sector and to skip the bad sectors. When OS/2 finds a bad sector, the sector is marked and OS/2 will no longer allocate your data to that sector.

RECOVER (recover files)

To recover a disk, enter the drive letter. For example, to recover all the files on drive A, enter:

```
RECOVER A:
```

Related Commands

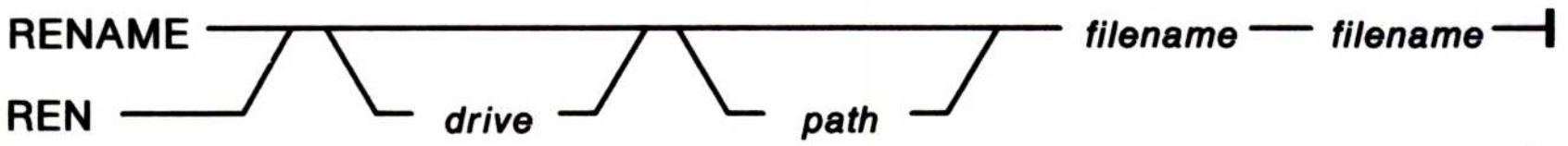
CHKDSK

RENAME (rename a file)

Purpose

Changes the name of a file.

Syntax



Remarks

You may type REN or RENAME. You must supply a drive name if the file is not on the disk in the default drive.

A path can be specified only with the first filename; therefore the file stays in the same directory after being renamed.

You may use global filename characters (* or ?) in either filename. All files matching the first filename are renamed. If global filename characters appear in the second filename, corresponding character positions are not changed.

Examples

The following example renames the file ABODE on drive C to PALACE.

```
RENAME C:ABODE PALACE
```

The following example renames the file ABODE on drive B to ABODE.XY.

```
REN B:ABODE *.XY
```

The following example renames the file MYPROG.COM in the directory \LEVEL1 on drive B to YOURPROG.COM.

```
REN B:\LEVEL1\MYPROG.COM YOURPROG.COM
```

Related Commands

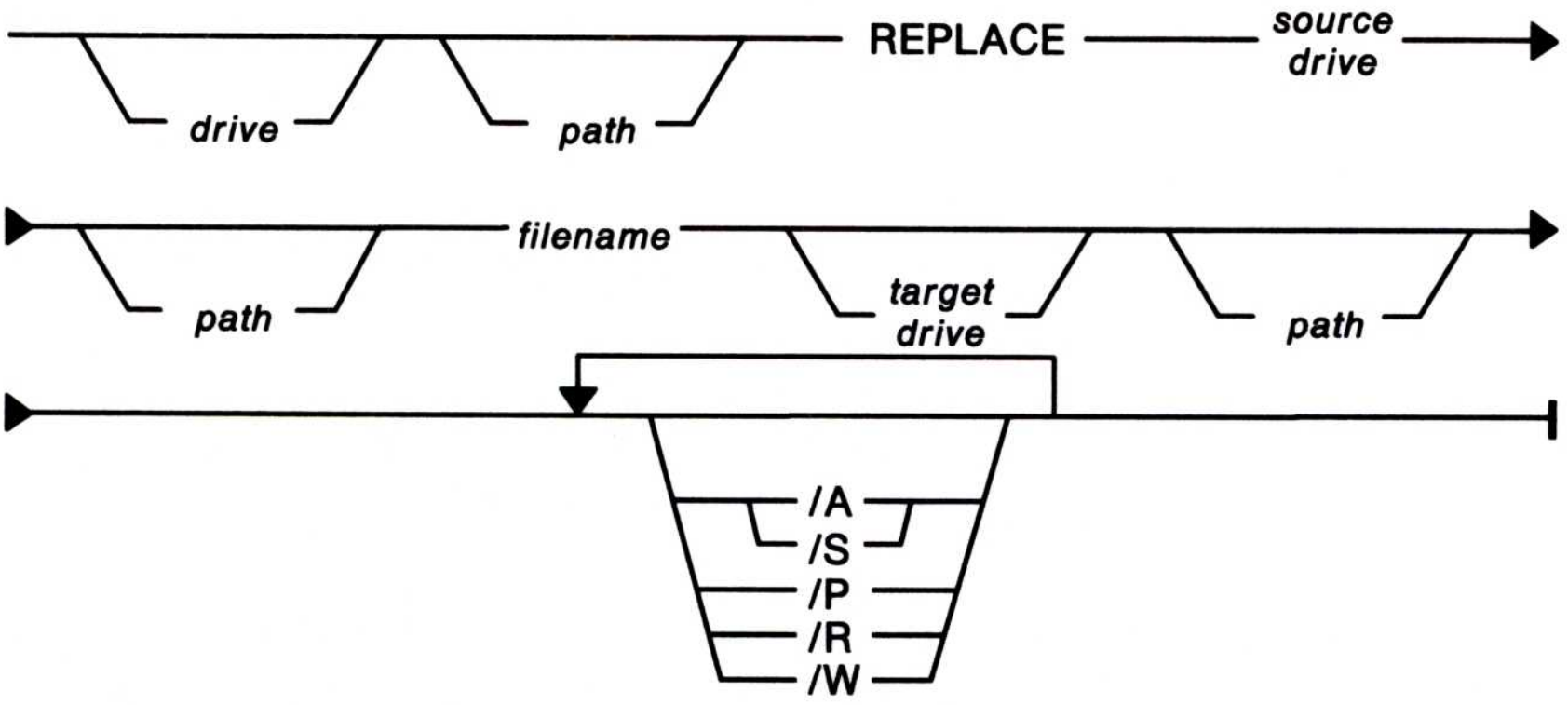
COPY

REPLACE (replace files)

Purpose

Selectively replaces files on the target drive with files of the same name from the source drive. Also selectively adds files from the source drive to the target drive.

Syntax



Parameters

/A Copies all files specified by the source that do not exist on the target. This allows you to add files to the target without overwriting the files that already exist on the target. You cannot use /A and /S together.

/S Searches all directories of the target drive for files matching the filename. You cannot use /A and /S together.

/P Prompts you as each file is encountered on the target, this allows selective replacing or adding.

/R Replaces files that are read-only on the target.

/W Waits for you to insert a diskette before beginning to search for source files.

If you did not specify /W with the command, REPLACE begins immediately.

Remarks

This is an external command. Hidden and system files are not found on the source and are not replaced on the target.

Example

Assume you have changed the file PE.PRO in your C:\EDITORS.DIR subdirectory. Now you want to selectively replace some of the other copies of PE.PRO on your fixed disk C. Enter the following:

```
REPLACE C:\EDITORS.DIR\PE.PRO C:\ /S /P
```

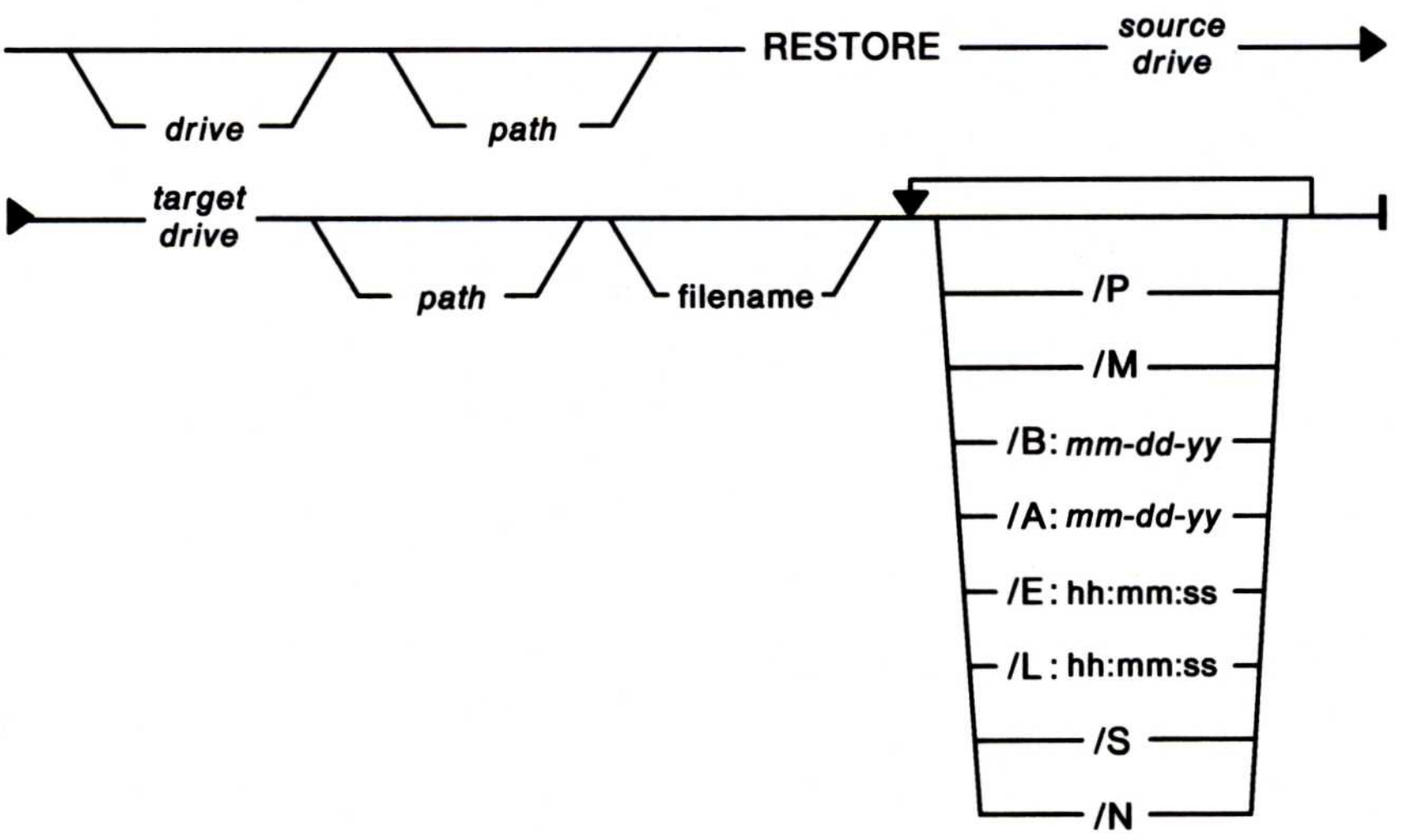
REPLACE then prompts you as each matching file is found on the target, allowing you to replace only those copies you choose.

RESTORE (restore backed up files)

Purpose

Restores one or more backup files from one disk to another.

Syntax



Parameters

/P Prompts for permission to restore any files on the *target disk* that match the file specification and are read-only or have changed since they were last backed up.

/M Restores those files on the *target disk* that have been modified since the last backup.

/B:*mm-dd-yy* Restores those files on the *target disk* last modified on or before the given date.

/A:*mm-dd-yy* Restores those files on the *target disk* last modified on or after the given date.

/E:*hh:mm:ss* Restores those files on the *target disk* last modified at or before the given time.

/L:*hh:mm:ss* Restores those files on the *target disk* last modified at or after the given time.

/S Restores any subdirectories from the *backup disk* to the target disk if they do not exist on the target disk.

/N Restores any files from the *backup disk* to the target disk if they do not exist on the target disk.

Note: Except for the /S and /N parameters, all other parameters are filter criteria that apply to the *target disk*. If the file does not exist on the target, the condition is always true.

Remarks

This is an external command that can restore files from disks of different media types, for example, from:

- Fixed disk to diskette
- Diskette to diskette
- Diskette to fixed disk
- Fixed disk to fixed disk

The source drive is the drive letter of the disk containing the files that were backed up. The target drive is the drive letter of the disk to which you want the files restored. You may optionally specify a path and filename after the target drive.

Note: The RESTORE command does not restore the following operating system files: IBMBIO.COM, IBMDOS.COM, COMMAND.COM, and CMD.EXE. Note that other system files are restored (depending upon filter criteria used) if they were backed up. Use the SYS command to restore the hidden files IBMBIO.COM and IBMDOS.COM. Use the COPY command to restore COMMAND.COM and CMD.EXE.

Examples

In the following example, all the files on the backup diskettes (including subdirectories) are restored to fixed disk drive C.

```
RESTORE A: C:\*.*  /S
```

This example restores files that have a filename extension of .DAT from backup diskette in drive A.

```
RESTORE A: C:*.DAT
```

RESTORE (restore backed up files)

This example restores three different files from the backup diskettes to the default fixed disk drive:

```
RESTORE A: \LEVEL1\FILE1.DAT
RESTORE A: \LEVEL1\LEVEL2\FILE2.DAT
RESTORE A: \LEVEL1\LEVEL3\FILE3.DAT
```

This example restores all files from the backup diskette in drive A, and prompts you if any files on drive A have changed since the last backup or if any files are marked read-only (/P).

```
RESTORE A: C: /P
```

Related Commands

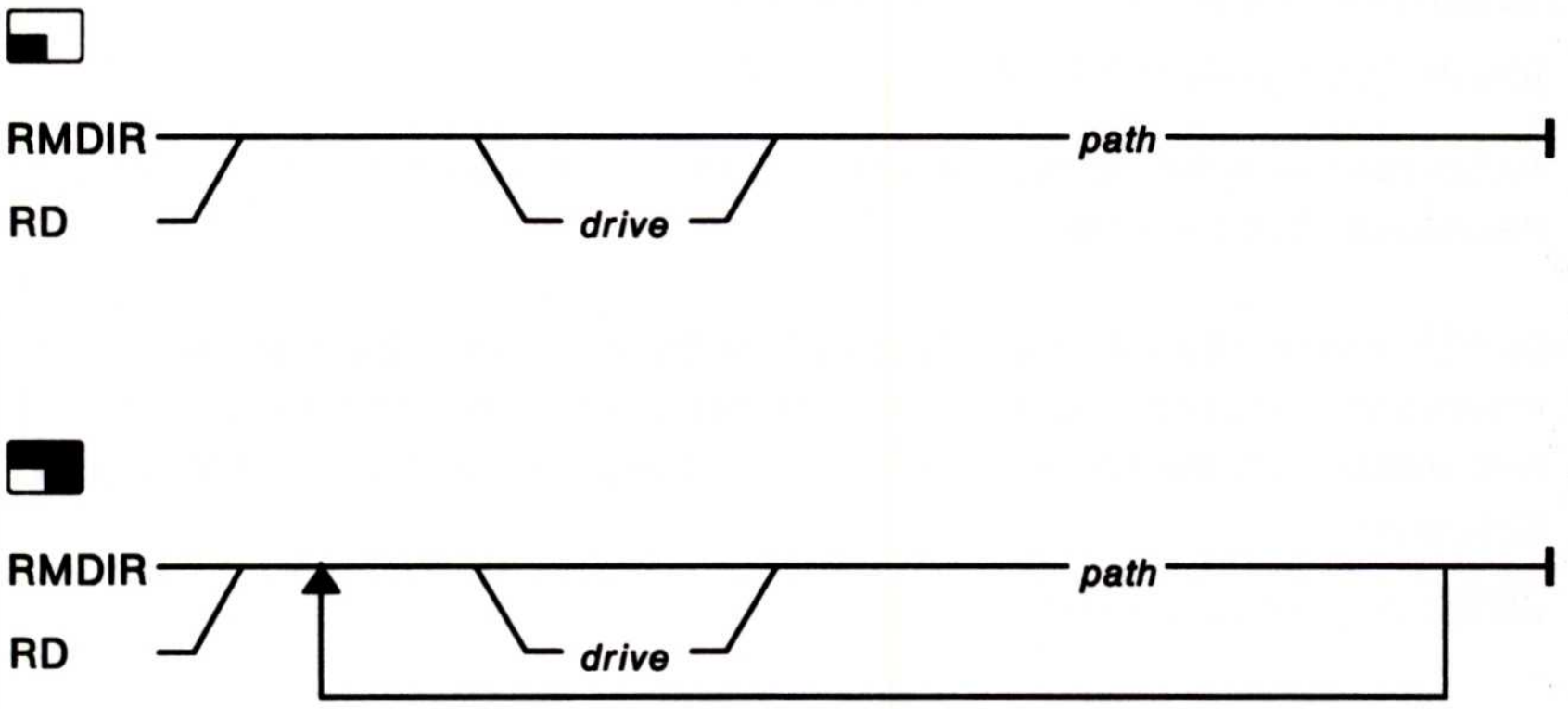
ATTRIB, BACKUP

RMDIR (remove directory)

Purpose

Removes empty directories from a multilevel directory structure.

Syntax



Remarks

You may type RMDIR or RD to remove a directory if you no longer need it. A directory can be removed only if it is empty. A directory is considered empty if it contains no hidden files and shows only the special entries (.) and (..). The (.) entry in a directory represents the current directory. The (..) entry in a directory represents the next level above the current directory.

The root directory and the current directory cannot be removed.

Note: If hidden files are in a directory, you cannot remove it. See the CHKDSK command on page 2-28 for determining the presence of hidden files.

DOS Mode Only: This command cannot be used on drives involved with the SUBST or JOIN command.

RMDIR (remove directory)

Examples

The following example removes the entry for LEVEL2 from the directory LEVEL1.

```
RD C:\LEVEL1\LEVEL2
```

In OS/2 mode, you can remove more than one directory at a time by specifying multiple paths, for example:

```
RMDIR \SALES\REPORTS \ACCTNG\PROGRAMS
```

removes the subdirectory REPORTS from SALES and the subdirectory PROGRAMS from ACCTNG.

In OS/2 mode you can specify multiple directories to be removed. This means that you can remove directories from different drives with only one RMDIR command. For example, at the OS/2 command prompt you can enter:

```
RMDIR A:\REPORTS C:\OS2\LETTERS
```

Related Commands

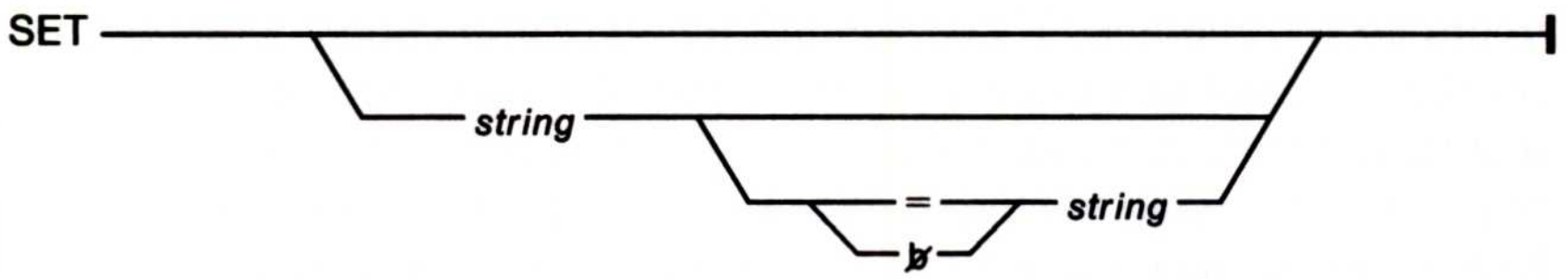
CHDIR, MKDIR, DIR, TREE

SET (environment variables)

Purpose

Sets one string in the environment equal to another string for later use in programs.

Syntax



Parameters

string The first *string* is the name of a system variable or replaceable parameter. The second *string* is the value you want to assign to the system variable or replaceable parameter.

Remarks

When a SET command is processed, if the name specified by the first string already exists in the environment, the command processor replaces its current value by the new value specified by the second string in the SET command. If you enter the SET command with only the first string, the command processor removes the environment variable or replaceable parameter name and its associated value from the environment, if the name exists. When you enter SET with no strings, the current environment settings are displayed.

Environments and Command Processors: The environment is a special place in storage that is used by the command processor to store and look up information. This information consists of values assigned to names. A name can be one that you create to use as a replaceable parameter, or it can be a system variable, such as PATH. Application programs (particularly compilers and assemblers) and batch files can use the information stored in the environment to affect their processing.

SET (environment variables)

Since the DOS session and OS/2 sessions operate independently from one another, each command processor that starts a session can have its own environment. For information on how command processors within a session inherit the environment, see the COMMAND command on page 2-34, and the CMD command on page 2-32.

Replaceable Parameters for Batch Files: The SET command can be used in batch processing to define a replaceable parameter as a name you create and then assign a value to. In the places where the value for the replaceable parameter name is to be substituted, you must begin and end the name with a percent sign. For example, suppose you have a batch file that contains multiple occurrences of a filename. To avoid having to change each occurrence when you want to change the name of the file, you can specify the filename as a replaceable parameter, for example:

```
%MYFILE%
```

Then to set %MYFILE% to a specific filename, you can include a SET command in your batch file, for example:

```
SET MYFILE=APRIL.RPT
```

After the batch processor interprets the SET command, it will change each occurrence of %MYFILE% it encounters in the batch file to APRIL.RPT.

You can also use names of system variables as replaceable parameters in batch files. Suppose you have the following replaceable parameter in a batch file:

```
%PATH%
```

When the batch processor encounters this replaceable parameter, it looks in the environment for the current value of PATH and substitutes it.

Commands that Are System Variables: Operating System/2 has system variables for which values can be set by entering their names as commands, or by specifying their names with the SET command. These system variables for OS/2 mode are:

PATH Sets a search path for executable files. (See the PATH command on page 2-101.)

DPATH Sets a search path for data files. (See the DPATH command on page 2-59.)

PROMPT Sets the system prompt. (See the PROMPT command on page 2-105.)

The commands that are system variables for DOS mode are the same with one exception: in DOS mode, the APPEND command sets the search path to data files. (See the APPEND command on page 2-8.)

Substituting Replaceable Parameters at the OS/2 Command Prompt: In OS/2 mode, substitution of replaceable parameter names by their values in the environment can be made at the command prompt. Using this feature of OS/2, you can append values to the values of system variables already in the environment. For example, suppose your current path to data files is:

```
DPATH C:\;C:\OS2;C:\OS2\INSTALL;
```

To append a path to the \MYDIR directory on drive C, you can enter:

```
SET DPATH=%DPATH%C:\MYDIR;
```

Or you can enter:

```
DPATH %DPATH%C:\MYDIR;
```

In both these examples, the C:\MYDIR directory is appended to the current path to data files in the environment of the current OS/2 session. In this example, it is assumed that your current path ends with a semicolon. If no environment variable exists by the name you specify, then no substitution is made.

Specifying System Variables with SET: Although you can use the SET command to set or replace the values for commands that are system variables, you should be aware that the syntax for SET differs from the system variable's command syntax.

SET (environment variables)

Following are examples using the PATH command that illustrate these differences.

Suppose you want to set the search path to executable files to null; that is, you no longer want to have a value for PATH in your environment. You can enter:

```
SET PATH=
```

This is equivalent to the PATH command's syntax:

```
PATH ;
```

However, if you confuse the syntax for PATH with the syntax for SET and enter:

```
SET PATH ;
```

this results in a syntax error and the path is not reset to null.

Be sure you specify a blank *or* or an equal sign, not both. For example, if you were to enter:

```
SET PATH =C:\APPS
```

This results in a variable being created whose name "PATH " ends with a blank character. This variable does not replace the path variable. The correct syntax is:

```
SET PATH=C:\APPS
```

To avoid possible syntax errors, consider using the SET command only when establishing new paths, not for displaying current paths or resetting the path to null.

Related Commands

APPEND, DPATH, PATH, PROMPT

SETCOM40 (set COM port address)

Purpose

Sets the COM port address so a DOS program can access the COM port interface directly to support a serial device, such as a plotter, printer, or mouse, when the COM0X.SYS device driver has been installed.

Warning: Do not use this command when the COM port specified with it can be accessed by currently running applications in OS/2 mode by means of the COM0XSYS device driver.

Syntax



Parameters

COM*n*	Is COM port COM1, COM2, or COM3
ON	Sets the port address for a serial device.
OFF	Removes the port address.

Remarks

SETCOM40 can be used to enable the COM port for a DOS application that otherwise cannot run properly when the COM0X.SYS device driver is installed. Such an application written for the IBM Personal Computer DOS environment supports a printer, plotter, mouse, or other serial device by going directly to the COM port interface, instead of going through the COM0X.SYS device driver the way programs written for the OS/2 multitasking environment do.

SETCOM40 (set COM port address)

Warning: Since OS/2 cannot manage programs using serial devices in DOS mode, managing these programs becomes your responsibility.

If an OS/2 program is currently using the COM port, or if an OS/2 program is queued to use it with the SPOOL command, you must allow these OS/2 mode programs to finish before attempting to access this COM port from DOS mode.

Suppose you want to run a DOS application that uses a serial device in DOS mode. You have an IBM Personal System/2 and your CONFIG.SYS file contains this statement that loads the COM device driver at system startup:

```
DEVICE=COM02.SYS
```

This statement loads the device driver for the PS/2, which supports serial ports COM1, COM2, and COM3. Let's assume that you want to use a serial printer that is attached to the COM1 port interface. Let's also assume that any applications running in OS/2 mode that access this interface by means of the COMO2.SYS device driver have finished, and you are at the DOS command prompt. You can then enter, for example:

```
SETCOM40 COM1=ON
```

This allows your program in DOS mode to find the address of the COM port interface, so output can be sent to your serial printer. Once you have issued SETCOM40, do not switch from DOS mode without first ending the DOS application that is using the COM port; doing so may cause it to fail.

Remember to remove the port address once your DOS application has finished (see the example below).

CAUTION:
Not all programs written for IBM Personal Computer DOS are suitable to use in DOS mode. Applications that have timing dependencies, such as those using communications via a modem or local area network, may fail to work correctly. Moreover, these failures may not be immediately apparent.

SETCOM40 (set COM port address)

For more information on running IBM DOS applications, see "Guidelines for Running DOS Programs" on page B-3.

Example

To remove the COM1 port address when your DOS application no longer needs it, enter:

```
SETCOM40 COM1=OFF
```

This allows OS/2 to again manage the port.

Related Commands

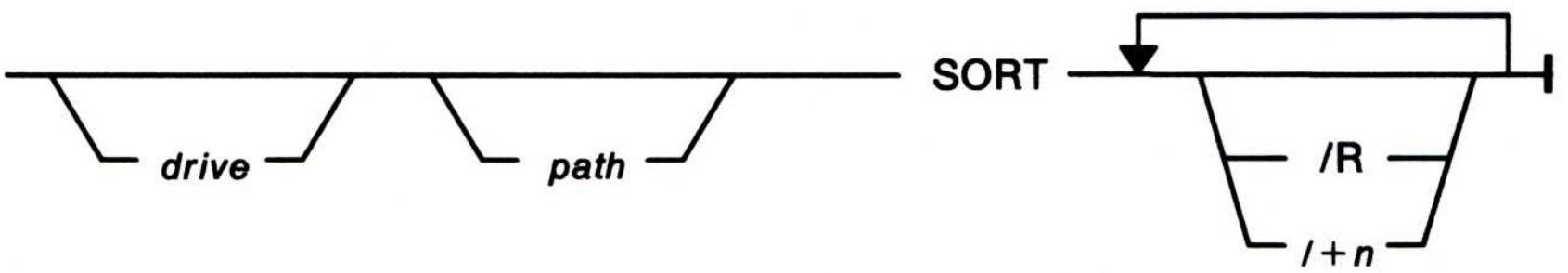
DEVICE=COM0X.SYS statement in CONFIG.SYS

SORT (filter)

Purpose

Reads data from standard input, sorts the data, then writes it to standard output.

Syntax



Parameters

/R	Reverses the sort; that is, sorts from Z to A.
/+n	Sorts starting with column *n*, where *n* is some number. If you do not specify this parameter, SORT begins sorting from column 1.

Remarks

SORT is an external filter. The standard input and output devices may not be redirected to the same filename. The maximum size of the input file is approximately 63Kb. SORT uses the collating sequence table, which identifies the default country code and selected code page, to perform the sort.

Characters are sorted according to their ASCII values, with the following exceptions:

- Lowercase letters a – z are equated to uppercase letters A – Z.
- Characters above ASCII value 127 are sorted according to rules based on the currently selected country code. For more information on country codes, see the COUNTRY statement for CONFIG.SYS on page 4-10.

Examples

The following example reads the file UNSORT.TXT, sorts it in reverse order, and then writes the output to the file SORT.TXT.

```
SORT /R <UNSORT.TXT >SORT.TXT
```

The following example pipes the output of the DIR command to the SORT filter. Then the directory listing is sorted starting with column 14 (column 14 contains the file size). The output is sent to the standard output device.

```
DIR ¦ SORT /+14
```

Related Commands

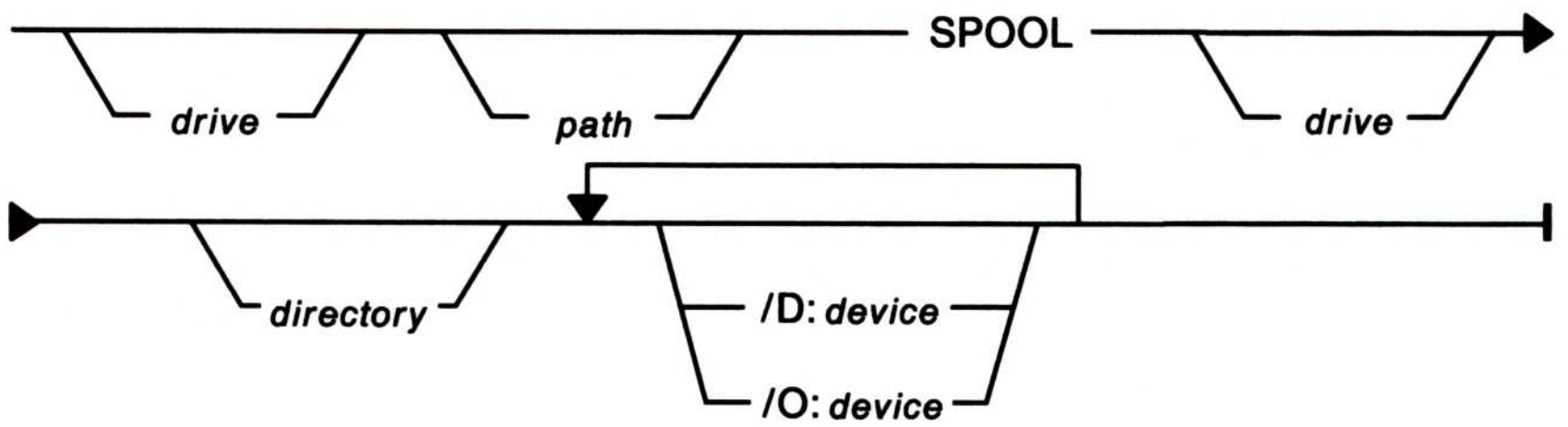
COUNTRY and CODEPAGE statements in CONFIG.SYS

SPOOL (print queue)

Purpose

Intercepts and separates data going to the printer from different sources so that printer output from concurrently running applications is not intermixed.

Syntax



Parameters

drive *****_directory_ Specifies the print spool subdirectory. This is where the temporary spool files will be created. The default is \SPOOL.

/D:*device* Specifies the input print device. This is the device the application program thinks it is printing to. Acceptable device names are LPT1, LPT2, LPT3, and PRN. If not specified, the default device is LPT1. COM*n* devices *cannot* be specified as input devices.

/O:*device* Specifies the output print device. This is the actual device on which output from the application program will be printed. If not specified, the default will be the same as the /D:*device*. COM1 through COM3 may be specified as output devices when a serial printer is attached to the serial port.

Note: The Base Asynchronous Communications Device Driver (COM0*x*.SYS) supports only COM1 through

COM3 for the IBM Personal System/2; and COM1 and COM2 for the IBM Personal Computer AT® and the IBM Personal Computer XT™. For the system to spool to a serial printer, it must load the COM0*x*.SYS device driver during system initialization. This requires a DEVICE statement in your CONFIG.SYS file. For more information on this device driver, see "COM0x.SYS Device Driver" on page 4-17.

Remarks

This external command can be used only if you have a printer attached to a parallel or serial port of your computer. The SPOOL command intercepts and sorts data being sent to printer device drivers by different processes so that the printer output is not intermixed. Data enroute to the printer is placed in temporary files on the disk in the subdirectory specified when starting SPOOL. Output is spooled until the application program instructs the print spooler to print the output.

Running SPOOL Noninteractively: Since there is no need for you to interact with the print spooler, it can process as a noninteractive program. System Installation does this for you by placing a RUN statement in your CONFIG.SYS file. It also creates the \SPOOL directory. When you start your system and initialization is completed, you receive a message informing you that the print spooler is running and that data will be printed on the printer specified by the RUN statement. For more information on the RUN statement, see page 4-51.

Printer Support: The print spooler supports only parallel devices (LPT1, LPT2, LPT3, and PRN) as input devices. SPOOL serves a single printer at a time.

Output devices can be parallel devices as well as serial devices COM1, COM2, and COM3. (Note that AUX is not an acceptable device name.) By specifying a parallel printer as the input device and a serial

---

AT is a registered trademark of the International Business Machines Corporation.

XT is a trademark of the International Business Machines Corporation.

printer as the output device, you can redirect parallel printer output to a serial device.

Note: If there is a DEVINFO statement for your printer and the print spooler is active for the printer, all output on that printer will default to the printer's Character Set 2. Using Character Set 2 is necessary to support printing in either of the two OS/2 code pages specified with the CODEPAGE statement in CONFIG.SYS. Character Set 1 can still be accessed using escape sequences. Refer to the instructions that came with your printer.

Warning: When spooling to a serial printer, the MODE command may have to be issued so that the COM*n* port interfaces correctly with the printer. However, if your printer is set to the defaults specified in the MODE command, you do not have to issue MODE. The MODE command can be issued before or after starting SPOOL. To find out how to query the status of the asynrchronous communications port, see the MODE command on page 2-88.

The print spooler will also support any other character device whose device driver contains monitor support compatible with the printer device driver.

The PRINT command interfaces to the SPOOL command when the PRINT command's /T or /C parameters are specified.

Supporting More Than One Printer: If you have a second parallel printer, you can run a second print spooler for it by placing another RUN statement in your CONFIG.SYS file:

```
RUN=C:\OS2\SPOOL.EXE /D:LPT2
```

In this example, the default *output* device is LPT2. Suppose you have a third printer, and it is attached to the COM2 serial port. Since this printer is a serial device, to have output printed on it, you must redirect input to it from a parallel port. You can do so by adding this statement to CONFIG.SYS:

```
RUN=C:\OS2\SPOOL.EXE /D:LPT3 /O:COM2
```

Note that you can specify LPT3 as the input device even though you do not have a parallel printer attached to this port.

You can also start another print spooler by using the DETACH command. For example, you can enter:

```
DETACH SPOOL /D:LPT3 /O:COM2
```

For more information on the DETACH command, see page 2-47.

Forcing Spooled Printer Output in DOS Mode: The print spooler spools output from a program into a file until it is instructed to close the file and print its contents. Many DOS programs do not contain the instructions required by the print spooler, so output is not printed until the application program ends.

If you wish to force printer output for a DOS program, you may do so while you are still interacting with the program by pressing Ctrl+Alt+PrtSc. This causes the end-of-file indicator to be sent to the file in the queue. Be sure that you wait until the entire file has been sent to the queue by your application program. Pressing Ctrl+Alt+PrtSc too soon can cause the print file to split into two separate print files.

Note: If you plan to use code page switching in DOS mode, you must start the print spooler before selecting the DOS command prompt from the Program Selector menu.

Switching Font Cartridges on Your Printer: If you are using code page switching on the IBM Quietwriter® III printer, there will be times when the correct cartridge is not in the slot. When this happens, the printer beeps, and the font light comes on. You must:

1. Insert the correct cartridge.
2. Press the Start button after the font light goes off.

Related Commands

MODE, PRINT, SETCOM40

Quietwriter is a registered trademark of the International Business Machines Corporation.

START (programs automatically)

Purpose

Starts an OS/2 mode program in another session.

Note: The primary use for START is to automatically start programs at system startup. The special batch file, STARTUP.CMD, allows you to do this.

Syntax



Parameters

string The title you want for your program, which will appear on the Program Selector menu. The string must be enclosed in double quotes and cannot exceed 30 bytes.

Note: The use of a DBCS double-byte character counts as two bytes. If a title is longer than 30 bytes, it is truncated.

If you do not specify *string*, the filename of your program appears on the Program Selector menu.

/C Ends the session when the *command* is complete. If the session created by the START command is displayed when the *command* ends, the Program Selector determines which session to display next. Usually, it is the Program Selector itself.

command An OS/2 internal or external command, a .CMD batch file, or any OS/2 program you want to pass to the command processor you are starting.

Remarks

If you enter START with no parameters, an OS/2 command processor is started and you are at the OS/2 command prompt.

START (programs automatically)

Why Use Start: Although START can be entered at the OS/2 command prompt, its primary purpose is to start OS/2 programs automatically at system startup. You can accomplish this by placing START commands in STARTUP.CMD, the OS/2 autostart facility. For examples and an explanation of the STARTUP.CMD special batch file, see "STARTUP.CMD File" on page 3-5.

Guidelines for Batch Files Run by START Commands: When you use the START command to run a batch file with the STARTUP.CMD file, your batch file should not make assumptions about the environment, current drive, or current directory.

For example, if the purpose of your batch file is to start a program on a specified drive within a specified directory, the complete name of the directory should be specified with the CD (change directory) command. Suppose your batch file starts the program called FILEMAN located in the \XYZ directory of drive D. Your batch file should look like this:

```
D:
CD \XYZ          Correct Batch File
FILEMAN
```

Do not *assume* the default for the current directory is the root directory, as in this example:

```
D:
CD XYZ          Incorrect Batch File
FILEMAN
```

Redirecting Input/Output: There are syntax considerations to be aware of when redirecting input or output with START. As an example, if you want to redirect output and you enter the following:

```
START "directory list to printer" DIR C:*.* > LPT1
```

CMD.EXE interprets this to mean to redirect the output of the START command to the printer. Therefore, to redirect I/O of the specified *command*, enclose the command string in quotes:

```
START "directory list to printer" "DIR C:*.* >LPT1"
```

Since the command string must be enclosed in quotes when redirecting I/O, the *program title* is a required argument; otherwise, START treats the command string as the program title.

START (programs automatically)



Example

If you enter:

```
START "Copy Files from A to B" XCOPY A:*.* B: /S
```

XCOPY is started in a new session, copying the files, including those in subdirectories, from diskette drive A to diskette drive B, and the program title will be displayed on the Program Selector menu.

Related Commands

CMD, STARTUP.CMD autostart batch file

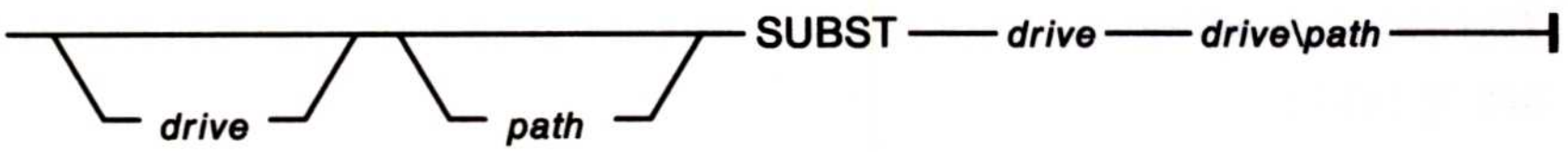
SUBST (substitute drives)

Purpose

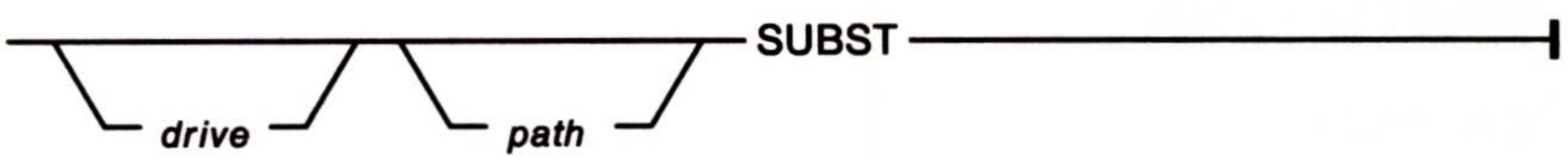
Substitutes a drive letter for another drive and path.

Syntax

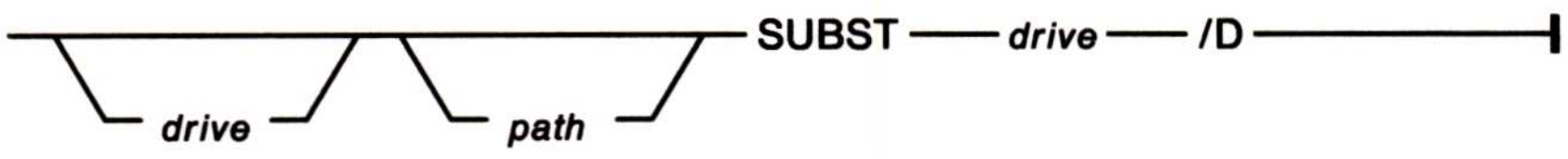
To substitute a drive letter for another drive and path:



To display substitutions currently in effect:



To delete substitutions currently in effect:



Parameters

drive drive path	The first drive is the drive letter you want to substitute for the drive and path that follow.
/D	Deletes a substituted drive and path

Remarks

SUBST is an external command that substitutes a drive letter for another drive and path so you can access that drive and path using only the drive letter.

When OS/2 finds a drive that was created with the SUBST command, it replaces the reference with the new path.

SUBST (substitute drives)



Examples

If you type:

```
SUBST
```

OS/2 displays the names of the substitutions currently in effect.

To substitute the drive letter G for the path C:\LEVEL\LEVEL1, enter:

```
SUBST G: C:\LEVEL\LEVEL1
```

You can now enter:

```
DIR G:FILE1
```

Instead of entering:

```
DIR C:\LEVEL\LEVEL1\FILE1
```

Related Commands

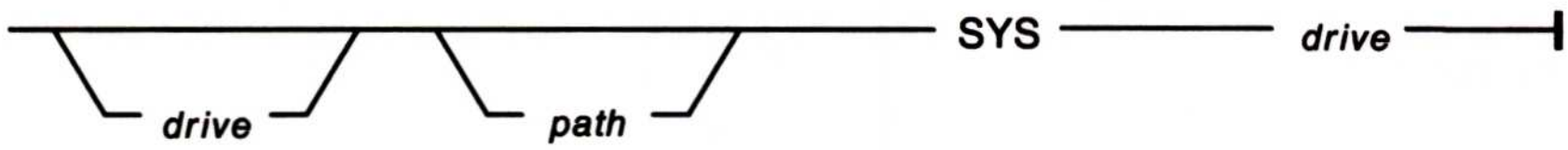
JOIN, ASSIGN

SYS (transfer hidden files)

Purpose

Transfers the OS/2 hidden system files IBMBIO.COM and IBMDOS.COM from the fixed disk or diskette in the default drive to the fixed disk or diskette in the specified drive.

Syntax



Remarks

SYS is an external command normally used to update the system or to place the hidden system files IBMBIO.COM and IBMDOS.COM on a formatted fixed disk or diskette. All files on the specified drive must be closed before running SYS. The specified drive cannot be the default drive.

The disk in the specified drive must be a newly formatted fixed disk or diskette or contain the system files IBMBIO.COM and IBMDOS.COM as the first two directory entries. Because Operating System/2 IBMBIO.COM is smaller than previous IBM Personal Computer DOS versions of IBMBIO.COM, you can transfer IBMBIO.COM and IBMDOS.COM from an OS/2 fixed disk or diskette to a DOS fixed disk or diskette without having to reformat.

The transferred files are copied in the following order:

IBMBIO.COM
IBMDOS.COM

IBMBIO.COM and IBMDOS.COM are both hidden files that do not appear when you type the DIR command. The DOS command processor COMMAND.COM, the OS/2 command processor CMD.EXE, and other system files specified in FORMATS.TBL required to start OS/2 are *not* transferred. You may use the COPY command to transfer them.

DOS Mode Only: This command does not recognize drive reassignments made with ASSIGN, JOIN, or SUBST commands.

SYS (transfer hidden files)

Example

You have a diskette in drive A that has IBM Personal Computer DOS Version 3.3 on it. You need to copy the OS/2 system files onto this diskette. The following command:

```
SYS A:
```

will do this for you.

Related Commands

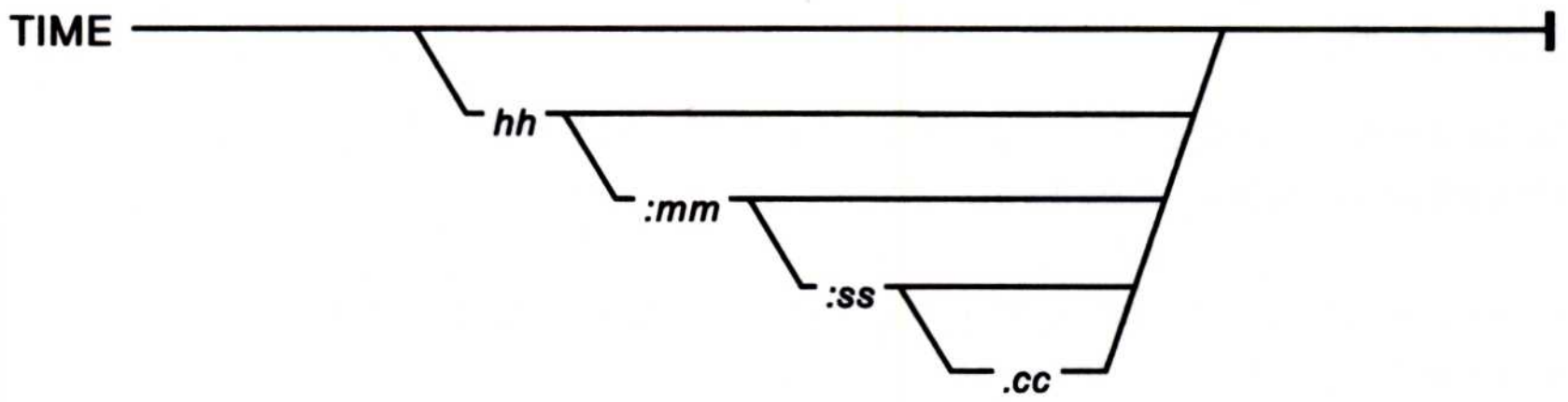
FORMAT

TIME (set system time)

Purpose

Displays or changes the time known to the system and resets the time on your computer's clock. This time is recorded in the directory when you create or change a file.

Syntax



Parameters

hh	The new hours must be entered using 0 – 24 only.
mm	The new minutes must be entered using 0 – 59 only.
ss	The new seconds must be entered using 0 – 59 only.
cc	The new hundredths of seconds must be entered using 0 – 99 only.

Separate the hours and minutes entries by a separator defined in the country dependent information file. For example, the U.S. separator is the colon.

You can specify *hh, hh:mm, hh:mm:ss,* or *hh:mm:ss.cc.* Values that are not specified are set to zero.

If you use *cc*, separate the hundredths of seconds with a decimal separator defined in the country dependent information file.

TIME (set system time)

Remarks

TIME is entered in a 24-hour clock format. If you just enter TIME, the following message is displayed:

```
The current time is: hh:mm:ss.cc
Enter the new time:_
```

Press the Enter key if you do not want to change the time shown. If you want to change the time after you have started OS/2, (for example, to 8:20 a.m.), enter:

```
TIME 8:20
```

in response to the OS/2 prompt. If you do not enter a valid time, OS/2 displays an error message.

Like the DATE command format, the TIME separators can be changed with the COUNTRY statement in the CONFIG.SYS file.

Example

Suppose you want to change the time to 21:58:00.00. First, enter:

```
TIME
```

Then when the prompt appears, you can enter the time you want:

```
The current time is: 13:55:49.65
Enter the new time: 21:58
```

Related Commands

COUNTRY statement in CONFIG.SYS, DATE, PROMPT

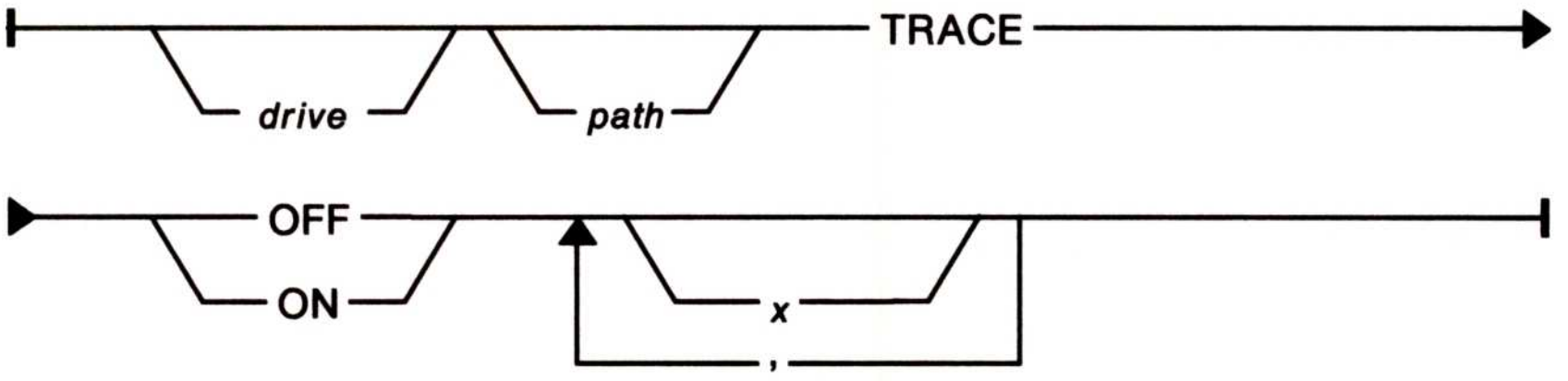
TRACE (system events)

Purpose

Selectively enables or disables the tracing of system events.

Note: This command is intended for use with aid from an IBM Service Representative. For more information, see "Commands for Problem Determination" on page 2-6.

Syntax



Parameters

ON	Permits system tracing of specified events.
OFF	Prevents system tracing of specified events.
x	A number between 0 and 255, indicating the major event code to be traced.

Remarks

TRACE is an external command. The value of TRACE on system startup can be defined in the CONFIG.SYS file. If no TRACE command appears in CONFIG.SYS, the default value of TRACE is OFF.

A major event code is the number assigned to a major system event, such as opening a file, writing to a file, or sending output to the display. Numbers for these major events will be supplied to you by your IBM Serice Representative.

If no events are listed after the ON or OFF parameter, then ON indicates all events will be traced and OFF indicates no events will be traced. If invalid event codes are selected, processing will continue on the remainder of the listed events but an error message is generated.

TRACE (system events)



For more information on TRACE, see the TRACE statement for CONFIG.SYS on page 4-59.

Related Commands

TRACE and TRACEBUF statements in CONFIG.SYS, TRACEFMT

TRACEFMT (trace formatter)

Purpose

Displays formatted trace records in reverse time stamp order.

Note: This command is intended for use with aid from an IBM Service Representative. For more information, see "Commands for Problem Determination" on page 2-6.

Syntax



Remarks

TRACEFMT is an external command that helps you to isolate causes of problems in the OS/2 system. The system trace formatter captures the current contents of the system trace buffer, analyzes each trace record, and sends the formatted data to standard output. This output may be redirected to a file or a printer.

TRACEFMT works only if you have the TRACE or the TRACEBUF statement in your CONFIG.SYS file. Either of these commands allocate a system trace buffer, and tracing is not required to be active (TRACE=ON) at the time TRACEFMT is issued.

The output generated by TRACEFMT will be analyzed by your IBM Service Representative. TRACEFMT can be used as many times as required to diagnose a problem without having to restart the system.

TRACEFMT (trace formatter)



Example

You have already entered a TRACE statement that has been tracing system events recommended by your IBM Service Representative. To format the trace records contained in the trace buffer and have the formatted output printed on your printer, enter the following at the OS/2 command prompt:

```
TRACEFMT >LPT1
```

Related Commands

TRACE and TRACEBUF statements in CONFIG.SYS, TRACE

TREE (display directory structure)

Purpose

Displays all of the directory paths found on the specified drive, and optionally lists the files in each subdirectory.

Syntax



Parameters

/F	Displays the names of files in the subdirectories.

Remarks

TREE is an external command that displays all of the directories on a disk. For each directory found, its full path is displayed, along with the names of any directories defined within it.

Examples

The following example lists all the directory paths and the names of all files in the root directory and each subdirectory on drive B. The output is placed in the file TREE.LST in the current directory of drive C.

```
TREE B:/F >C:TREE.LST
```

The following example lists all the directory paths and filenames of the root directory and of all subdirectories on drive A. The output goes to the printer.

```
TREE A:/F >PRN
```

Related Commands

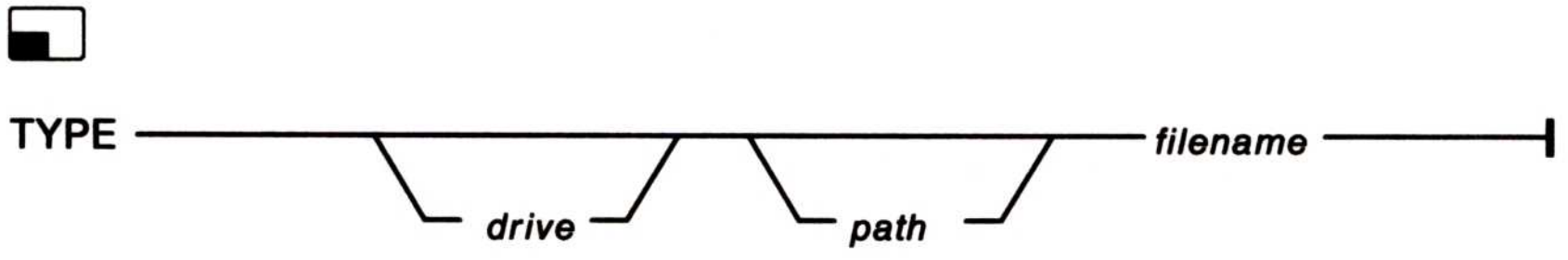
DIR, CHDIR, MKDIR, RMDIR

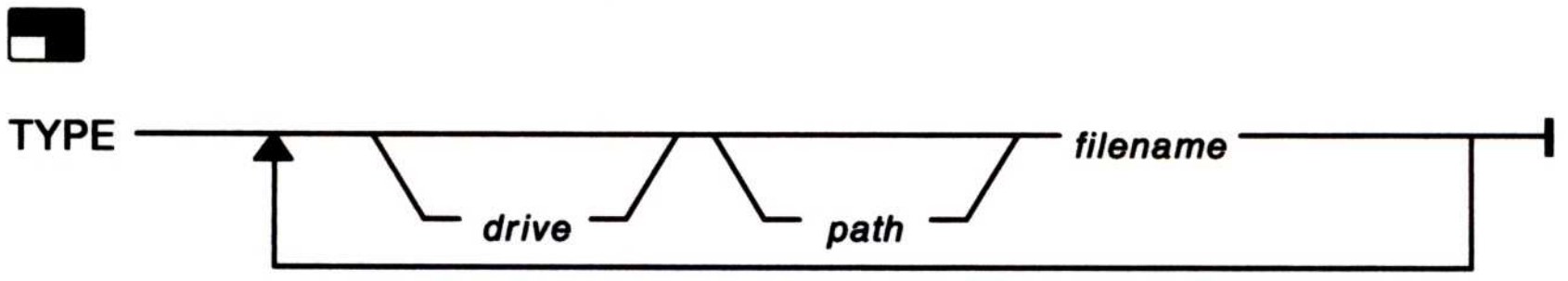
TYPE (display file contents)

Purpose

Displays the contents of a file on the screen.

Syntax





Remarks

You can use this command to view a file without modifying it. When you use TYPE to display a file with tabs in it, each tab character is expanded to eight spaces. A display of binary files causes control characters (such as Ctrl+Z) to be sent to your computer, including bells, form feeds, and escape sequences.

OS/2 Mode only: TYPE displays files consecutively on the screen if you specify multiple filenames. You can include *drive* and *path* when specifying multiple filenames. You may also use global filename characters * and ? in the filename you specify.

Examples

The following example entered at the OS/2 command prompt displays the files MYFILE1.TXT on drive C and MYFILE2.TXT on drive A on the standard output device.

```
TYPE C:MYFILE1.TXT A:MYFILE2.TXT
```

TYPE (display file contents)

Suppose you have the filenames CALC.BAT and CALC.BAK in your current directory. If they are the only files named CALC, and you enter the following at the OS/2 command prompt:

```
TYPE CALC.BA?
```

this produces the same results as entering:

```
TYPE CALC.BAT CALC.BAK
```

The contents of both files are displayed.

Suppose you want to pause the display of a long file so that you can read it. You can use the MORE command with TYPE to do this.

```
TYPE MYFILE.NEW | MORE
```

For more information on the MORE command, see page 2-97.

Related Commands

MORE

VER (display OS/2 version number)

Purpose

Displays the OS/2 version number.

Syntax

Example

If you enter the following at either the OS/2 or DOS prompt:

```
VER
```

the version number for OS/2 is displayed:

```
The IBM Operating System/2 Version is 1.00
```

Related Commands

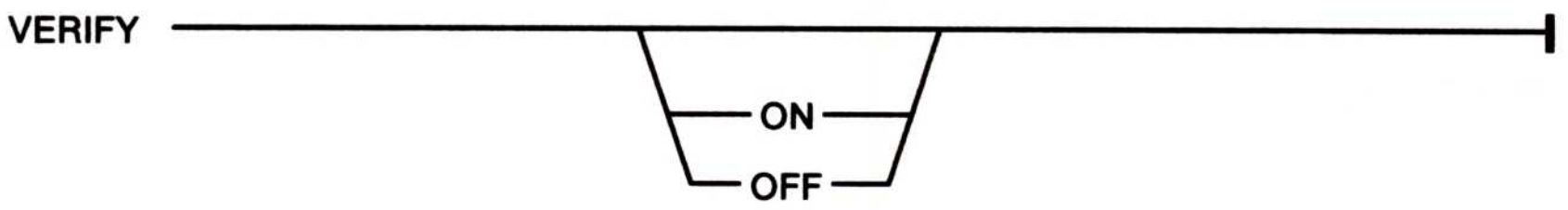
PROMPT

VERIFY (set data verification)

Purpose

Confirms that data written to a disk has been correctly written.

Syntax



Parameters

ON	Verifies that your files are written correctly to the disk.
OFF	Turns verification off. This is the default value.

Remarks

When you issue:

```
VERIFY=ON
```

verification is done for file system I/O writes for both fixed disks and diskettes on a per session basis. See the MODE diskette verification command on page 2-96.

You can use the VERIFY command to tell OS/2 to verify that files are written correctly to the disk (no bad sectors, for example). OS/2 will perform a VERIFY each time you write data to a disk. You will receive an error message only if OS/2 is unable to write the data to the disk successfully. This command has the same purpose as the /V parameter in the COPY and XCOPY commands.

VERIFY ON remains in effect until you change it in a program by a SET VERIFY system call, or until you enter VERIFY OFF.

VERIFY (set data verification)

Example

If you want to know what the current setting of VERIFY is, enter:

```
VERIFY
```

with no parameter.

Related Commands

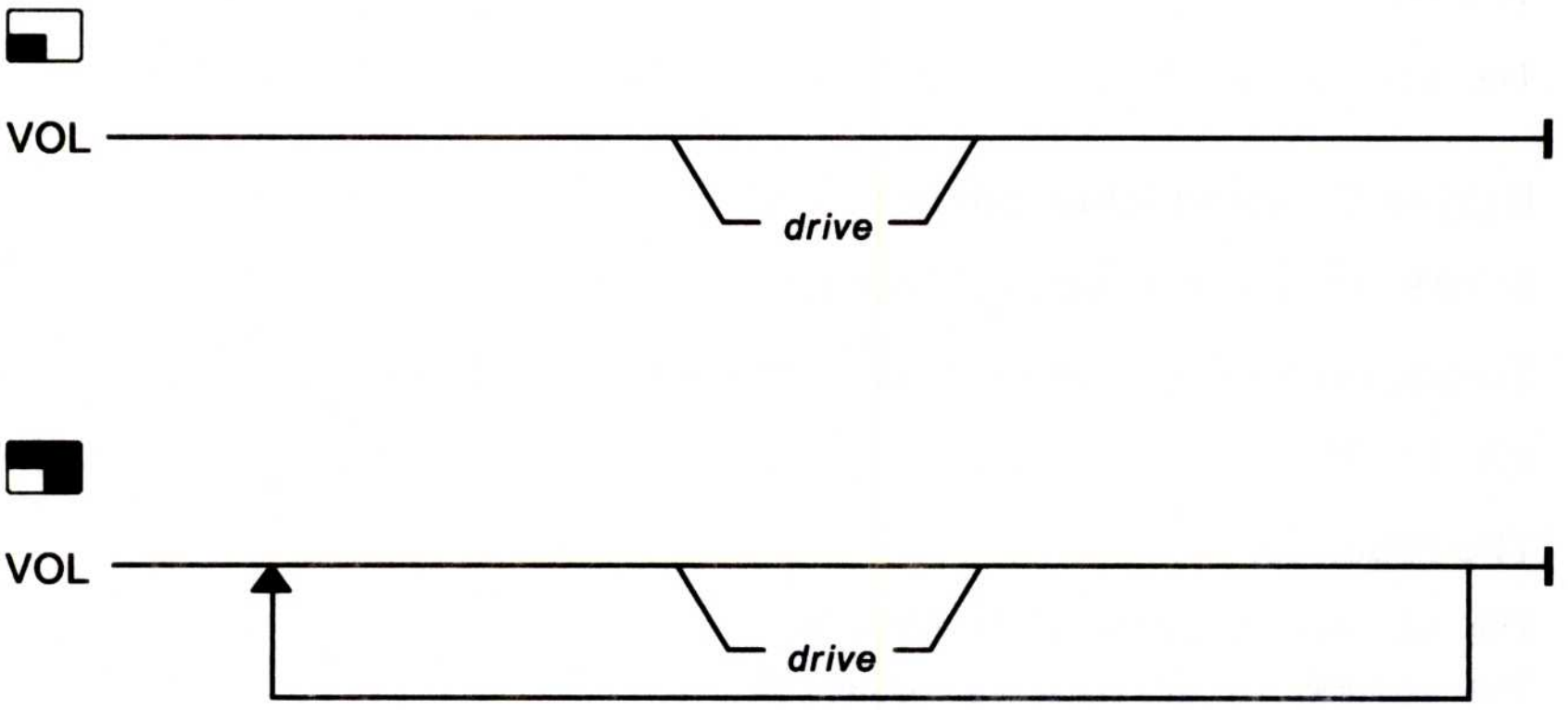
MODE DSKT

VOL (display disk volume label)

Purpose

Displays the disk volume label if it exists.

Syntax



Remarks

This command displays the volume label of the disk in a specific drive. If you do not enter a drive letter, OS/2 prints the volume label of the disk in the default drive.

In OS/2 mode, if you specify more than one drive, VOL displays the volume labels consecutively.

VOL (display disk volume label)

Examples

The following examples show you what is displayed when you are at the OS/2 command prompt. To display the volume label of drive C, enter:

```
VOL C:
```

The result is:

```
The volume in drive C is MYDISK.
```

If drive C has no label, the result is:

```
Volume in drive C has no label.
```

To display the volume labels of more than one drive, enter:

```
VOL C: D:
```

The result is:

```
The volume in drive C is MYDISK.
The volume in drive D is YOURDISK.
```

Related Commands

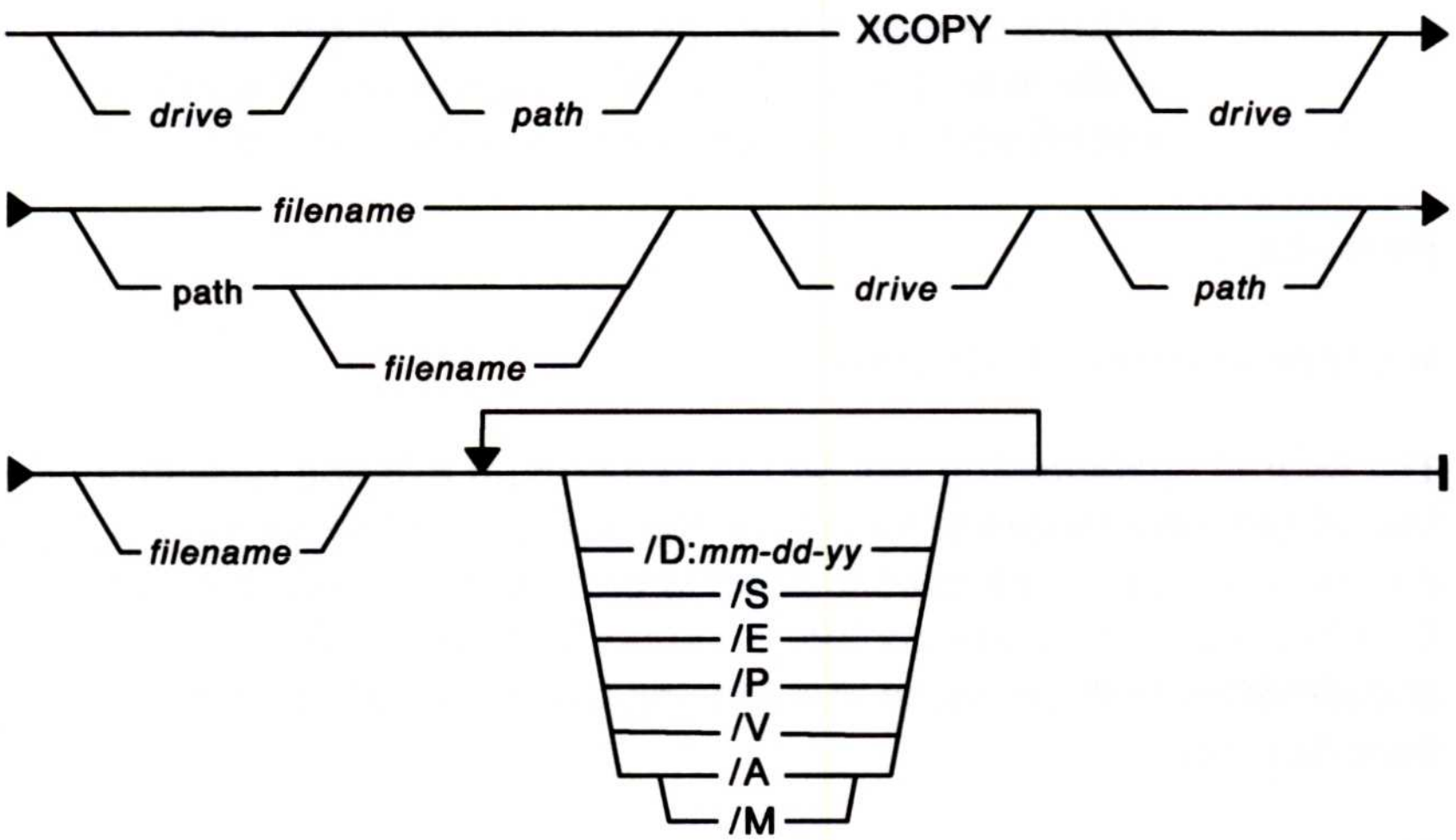
FORMAT, LABEL

XCOPY (copy subdirectories)

Purpose

Selectively copies groups of files, which can include lower level sub-directories.

Syntax



Parameters

/S Copies subdirectories. Without this parameter, XCOPY works only within a directory. This parameter does not create the corresponding source directory on the target if no files have been found on the source.

/D:*mm-dd-yy* Copies files that have been modified on or after the specified date.

/E When used together with the /S parameter, this copies the entire tree structure while creating corresponding source directories on the target even if no files have been found on those source directories.

/P Prompts you before performing a physical copy. You will respond to a (Y/N)? prompt for a selective copy. With this parameter, XCOPY copies on a file-by-file basis.

XCOPY (copy subdirectories)

/V Verifies that data written to a disk has been correctly written.

/A Copies archived files only. This parameter does not turn off the archive bit of the source file.

/M Copies archived files only. Unlike /A, this parameter turns off the archive bit of the source file. So when you regularly back up the source disk with XCOPY, using this parameter increases the efficiency of copying. You should be aware that the BACKUP command uses the archive bit also. If both the /A and /M parameters are specified, the one specified last is the one used.

Remarks

XCOPY is an external OS/2 command.

The source specification can be the name of an existing directory or file. XCOPY first checks to see if the specification is the name of a directory or file. If the specification is the name of a directory, all the files in the directory are copied to the target specification. If the specification is is the name of a file, the file is copied to the target specification.

The target specification can be the name of an existing directory or file, or it can be the name of a new directory or file you want to create.

If the source specification is the name of an existing directory, and the target specification is also the name of an existing directory, all the files in the source directory are merged into the target directory. If the source specification is a directory and the target specification is the name of a new directory you want to create, XCOPY prompts you to verify that it is a directory by asking:

```
Is the target specification a directory (Y/N)?
```

If you answer **Y**, a new directory is created on the target that contains copies of all the files in the source directory. If you answer **N** to this query, XCOPY treats the target specification as a filename.

If either specification does not contain a path, the current directory is assumed.

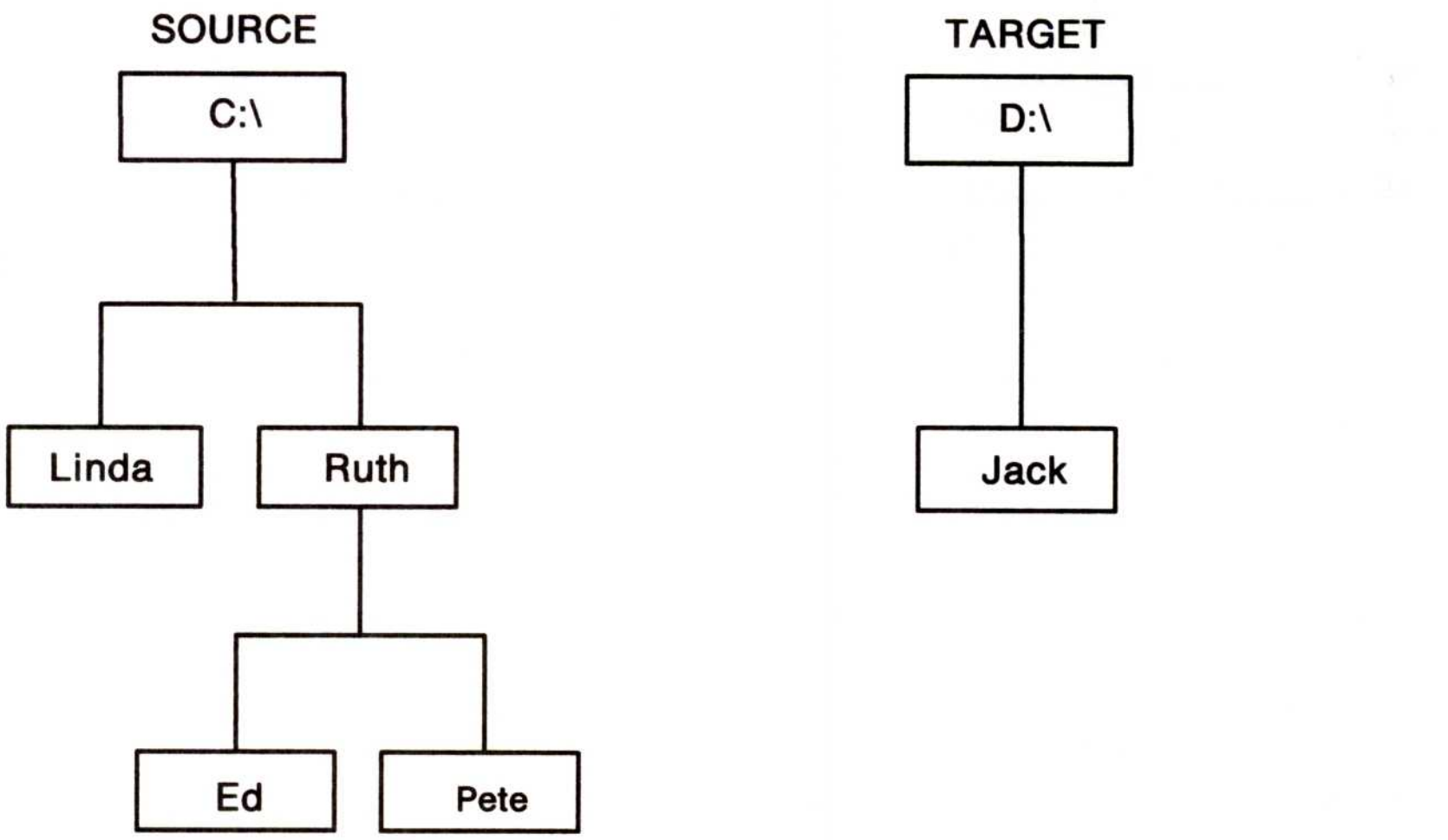
XCOPY (copy subdirectories)

Comparing COPY and XCOPY function: COPY A:\FILE1 B:\ copies the FILE1 from the source and creates a file FILE1 in the target root directory, while in the XCOPY case, XCOPY A:\DIR1 B:\ does not create a directory DIR1 in the target, nor does it try to merge into DIR1 if DIR1 already exists in the target. Instead, all the files in the starting source directory DIR1 are merged into the root directory of the target B (Remember the default filename is *.*.) This is because the COPY command performs its action at a given fixed directory boundary and, XCOPY, at a given tree structure boundary.

As the syntax shows, you can also specify the target filename just like the COPY command.

Examples

Suppose you have the following tree structures, where "C" is the source drive and "D" is the target drive:

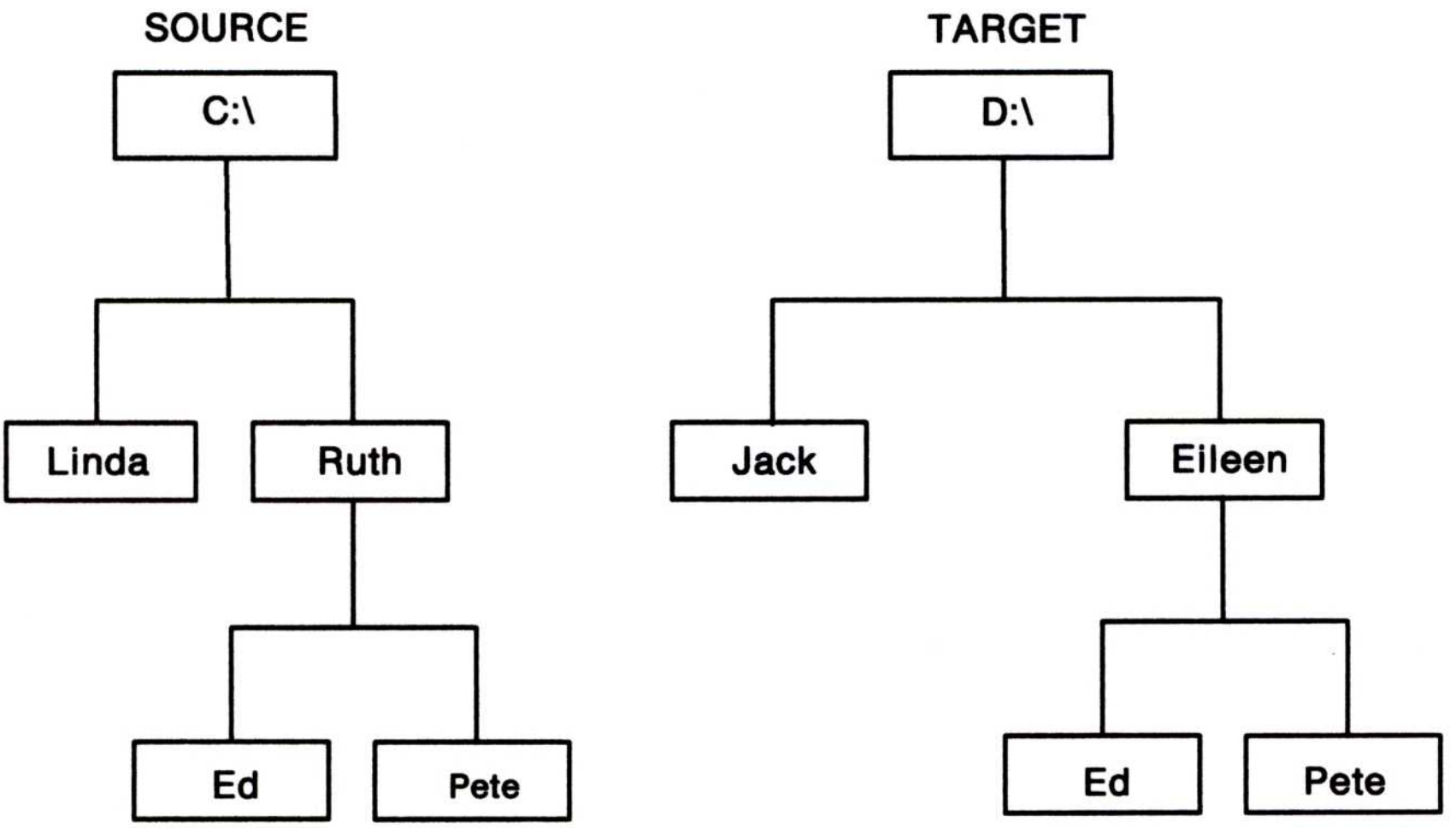


XCOPY (copy subdirectories)

If you enter:

```
XCOPY C:\RUTH D:\EILEEN /S
```

The result, as shown in the following illustration, is a new directory called EILEEN created on drive D that contains copies of all the files in RUTH, with the subdirectories ED and PETE copied with their contents beneath EILEEN.

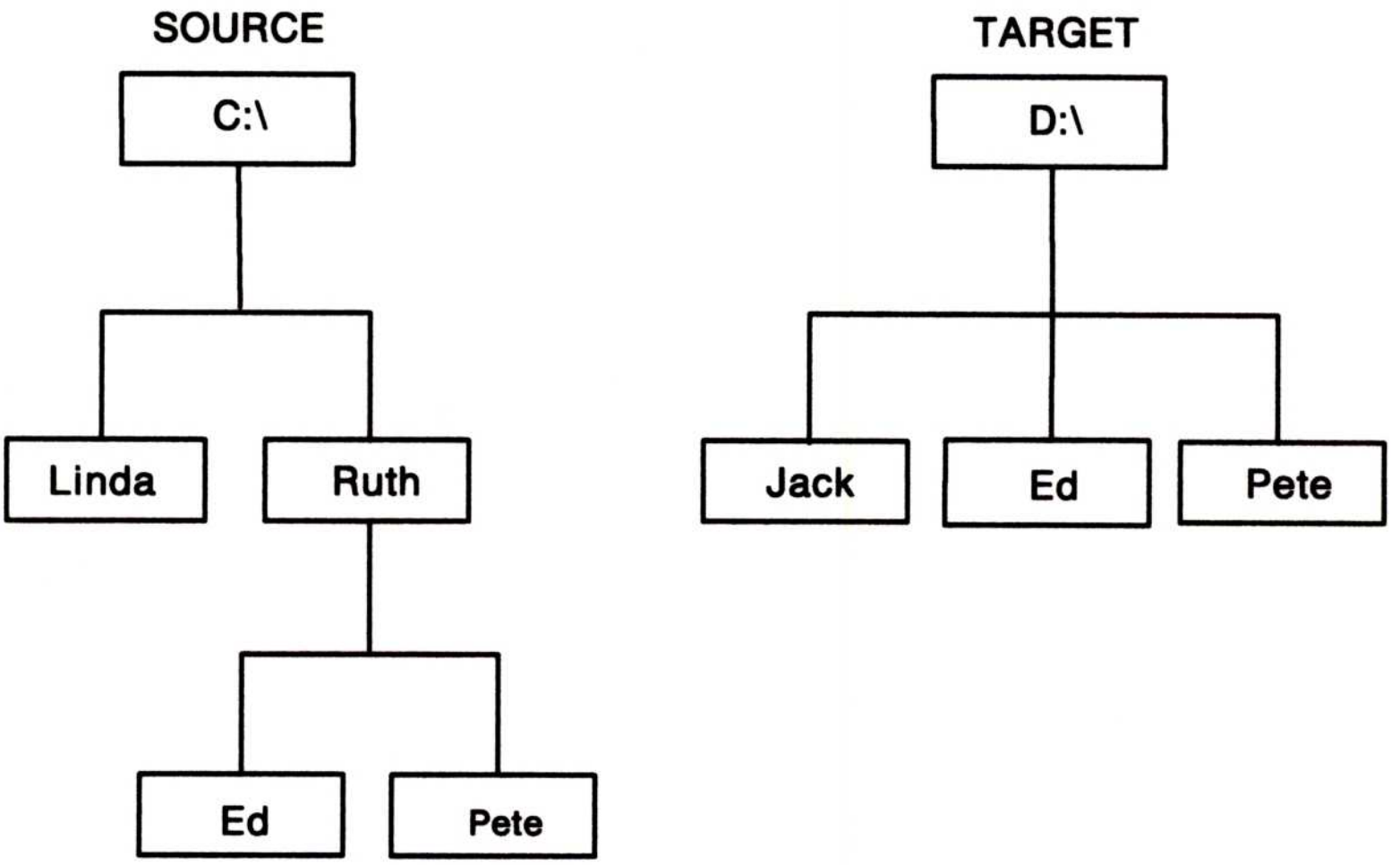


XCOPY (copy subdirectories)

Suppose, given the original source and target drives, you enter:

```
XCOPY C:\RUTH D:\ /S
```

The result is:



In this case, the files in RUTH are merged into the root directory on drive D, with the subdirectories ED and PETE copied with their contents beneath the root directory.

Related Commands

BACKUP, COPY, VERIFY

Chapter 3. Commands for Batch Files

Operating System/2™ (OS/2™) has a set of batch commands intended for use only in batch files. These commands appear in alphabetic order starting on page 3-11 of this chapter. For an overview of batch commands and their functions, see "Batch Commands" on page A-6.

Using Batch Files

Batch files can assist you in performing repetitive tasks. Like an OS/2 command, a batch file is run from the command prompt when you enter its name. You can include regular OS/2 commands as well as the special batch commands in a batch file. These commands are processed in the order they appear in the batch file.

Like any other filename, a batch filename follows the format of 1 to 8 characters followed by an extension. Although you need not include the batch file's extension when you enter its name at the command prompt, the extension is necessary to identify the file as a batch file to the command processor. There are two different extensions for batch files:

Mode	Extension
DOS	*filename*.BAT
OS/2	*filename*.CMD

Since some OS/2 commands run only in OS/2 mode and some run only in DOS mode, different extensions can help you avoid running a batch file that contains these commands in the wrong mode. A batch file with an extension of .BAT will not run in in an OS/2 session, and a batch file with an extension of .CMD will not run in the DOS session.

Warning: The different batch file extensions for DOS and OS/2 do not prevent you from putting a command that works only in one mode into a batch file whose extension is for the other mode. If you want to use a command in a batch file and you are not sure whether it works in both modes, refer to the command description before you use it.

Creating Batch Files

You can create a batch file or change an existing one by using EDLIN in DOS mode (see Appendix C, "The Line Editor (EDLIN)" on page C-1). You can also create a batch file by using the COPY command directly from the keyboard.

Step-By-Step Instructions

Follow these steps to create a batch file using the COPY command:

1. At either the OS/2 or the DOS prompt, enter:

   ```
   COPY CON filename.BAT
   ```

   Substitute the name you want for your DOS batch file for *filename*. (Of course, if you were creating a batch file to run in OS/2 mode, your batch file extension would be CMD.)

2. Enter the commands you want to include in the batch file.

   Note: If you make a typing mistake, press Ctrl+Break. This ends the COPY command *without* saving the batch file. Then repeat the procedure again.

3. Press F6 or Ctrl+Z and then Enter when you finish typing commands. This ends the COPY command and saves the batch file. A message appears, saying the file has been copied.

Running a Batch File

You can run batch files by typing the name of the batch file at the command prompt and then pressing Enter. Entering the batch filename extension .BAT or .CMD is optional. For example, to run a batch file named EXAMPLE.CMD, you can enter the following at the OS/2 command prompt:

```
EXAMPLE
```

If you have two batch files with the same filename but different extensions for the two modes, the command processor will run the one with the appropriate extension.

You can make batch files run batch files for you. There are two ways you can do this:

1. By *chaining* batch files. Make the last command in a batch file the name of another batch file you want to start after the current one ends.
2. By *nesting* batch files. Use the CALL command in a batch file to call another batch file. The entire sequence of batch file statements is processed as a single command.

To stop the running of a batch file, press Ctrl+Break.

Batch Processing Differences

Batch file processing in DOS mode is very similar to IBM Personal Computer DOS. If a *filename*.BAT is running in the DOS session and you press Ctrl+Break, the following prompt is displayed:

```
Terminate batch job (Y/N)?
```

You can enter Y in response to this prompt, which causes the batch file to end and control to be returned to the operating system. You can also enter N, and the batch file will resume its processing.

However, pressing Ctrl+Break in a OS/2 session does not give you this choice. Once you press Ctrl+Break to stop a *filename*.CMD from running, it ends and control is returned to the operating system.

Treating batch files dependent on others as a single command avoids unpredictable results in a multitasking environment. For example, suppose a batch file that sets environment variables with SETLOCAL is interrupted. Any environment values it may have set with SETLOCAL are no longer in effect. If the batch file is then allowed to resume processing, the results may be very different from what was intended.

Special Batch Files

During system installation two batch files are created for you:

- OS2INIT.CMD
- AUTOEXEC.BAT.

These files are placed in the root directory of the startup drive. You can change them the way you would any other batch file by using an editor in DOS mode.

You can also create a batch file called STARTUP.CMD, if you want to use the OS/2 autostart facility.

OS2INIT.CMD File

Every time you select the "OS/2 Command Prompt" from the Program Selector menu to start an OS/2 session, the system looks for a batch file named OS2INIT.CMD in the root directory of the startup drive. This support is turned on or off by the PROTSHELL statement in CONFIG.SYS. For more information on the PROTSHELL statement, see page 4-47.

OS/2 creates an OS2INIT.CMD file for you during system installation that contains these commands:

```
PATH C:\;C:\OS2;C:\OS2\INSTALL;
DPATH C:\;C:\OS2;C:\OS2\INSTALL;
CALL HELP ON
```

PATH and DPATH provide search paths to OS/2 external commands and data files. You can modify these commands so that they include the paths to your application programs and any data files they require.

The CALL command runs a batch file that turns on the HELP bar as part of your OS/2 command prompt. This batch file includes the following prompt command:

```
PROMPT $P$G
```

If you include another prompt command after CALL HELP ON, it will override the HELP prompt.

AUTOEXEC.BAT File



The first time you select the DOS command prompt from the Program Selector, the DOS command processor searches for a special batch file named AUTOEXEC.BAT in the root directory of the startup drive. An AUTOEXEC.BAT file is useful if you want to run certain commands when you start DOS. (AUTOEXEC.BAT does not run in OS/2 mode.)

OS/2 creates an AUTOEXEC.BAT file during the installation process. It contains these commands:

```
PATH C:\;C:\OS2;
CALL HELP ON
```

The PATH command provides a search path to OS/2 external commands. You can add your own paths to application programs, and also include an APPEND command to specify paths to data files. For more information on APPEND, see page 2-8.

The CALL command runs a batch file that turns on the HELP bar as part of your OS/2 command prompt. This batch file includes the following prompt command:

```
PROMPT $P$G
```

If you include another prompt command after CALL HELP ON, it will override the HELP prompt.

You are not prompted for the date and time when you start the system. You are also not prompted when you select the DOS command prompt from the Program Selector, providing an AUTOEXEC.BAT file exists. If an AUTOEXEC.BAT file does not exist, you are then prompted for the date and time. Since the system uses the clock in your computer, it is not necessary to set the time and date each time you start the system or start DOS.

STARTUP.CMD File



You can take advantage of the OS/2 autostart facility by creating a special batch file called STARTUP.CMD. When you start OS/2, the first OS/2 session is started automatically and switched to the screen if there is a special batch file named STARTUP.CMD in the root directory of

the startup drive. If there is no STARTUP.CMD file, the program selector menu is displayed.

You can use the internal command START in the STARTUP.CMD file to start programs in additional OS/2 sessions.

Example 1: Suppose you have the following statements in your STARTUP.CMD file:

```
PATH C:\;C:\OS2;\MISCPGM;
START CALC
START WP
START
CALENDAR
```

This is what happens when you start OS/2:

1. STARTUP.CMD starts a command processor in a session that is displayed on your screen. This is where the statements in STARTUP.CMD will be processed.
2. The path is set for the first session.
3. The first START command starts a second session that runs a spread sheet program.
4. The second START command starts a third session that runs a word processing program.
5. The third START command starts a command processor in a fourth session. (When you switch to the Program Selector, CMD.EXE appears on the "Switch to a running program" list.)
6. The last statement runs the CALENDAR program in the first session.

Note: Environment variables defined in STARTUP.CMD are inherited only by programs that run in the first session. Any other sessions started with START commands are not affected. Thus, the path specified in the example above is inherited only by programs that run in the session where the CALENDAR program is running.

Example 2: Suppose you had these statements in STARTUP.CMD:

```
START CALC
START WP
```

When you start the system, the sessions for the spread sheet and word processing applications are started. The first session started by STARTUP.CMD displays the command prompt on your screen.

Example 3: When this STARTUP.CMD file runs:

```
START CALC
START WP
EXIT
```

the sessions for the spread sheet and word processing applications are started, and the EXIT command causes the command processor started in the first session by STARTUP.CMD to end. The Program Selector menu is then displayed.

A program specified with a START command can be an internal or external command, an application program, or a batch file. If a batch file is started that has as its last statement the EXIT command, when the batch file ends, the command processor is exited and the session is ended.

Guidelines for Batch Files Run by START Commands: You can set environment variables for a session by having a START command in STARTUP.CMD run a batch file. Here is an example of what the batch file might contain:

```
PATH C:\;C:\OS2;C:\MISCPGM;
DPATH C:\;C:\OS2;C:\LETTERS;C:\REPORTS;
PROMPT [EDIT]
EDIT
```

The last command in this example starts an editing application. When you end the application, the command prompt appears as:

```
[EDIT]
```

When you use the START command in STARTUP.CMD to run a batch file, your batch file should not make assumptions about the environment, current drive, or current directory.

For example, if the purpose of your batch file is to start a program on a specified drive within a specified directory, the complete name of the directory should be specified with the CD (change directory) command. Suppose your batch file starts the program called FILEMAN located in the \XYZ directory of drive D. Your batch file should look like this:

```
D:
CD \XYZ          Correct Batch File
Batch File
FILEMAN
```

Do not *assume* the default for the current directory is the root directory as in this example:

```
D:
CD XYZ          Incorrect Batch File
FILEMAN
```

A complete description of the START command is on page 2-130.

Using Replaceable Parameters in a Batch File

By using replaceable parameters, you can pass data to a batch file when it runs. The file can do similar work with different data each time you run it. These replaceable or "dummy" parameters, named %0 – %9, are place holders for the values that you supply when you enter the batch command from the OS/2 command prompt or from another batch file. For example, if the OS/2 batch file EXAMPLE.CMD contains these commands:

```
TYPE %0.CMD
COPY %1.TAB %2.TAB
TYPE %2.PRN
```

OS/2 sequentially replaces %1, %2, %3, and so on up to %9 with the parameters supplied when the file is run. The parameter %0 is always replaced by the drive and path, if specified, and the filename of the batch file.

To run the EXAMPLE.CMD file and pass parameters, enter the batch filename at the OS/2 prompt followed by the parameters you want sequentially substituted for *%1*, *%2*, and so on.

For example, you can enter:

```
EXAMPLE C:PROG1 B:PROG2
```

EXAMPLE is substituted for %0, C:PROG1 for %1, and B:PROG2 for %2.

The result is the same as if you entered each of the three commands (in the EXAMPLE.CMD file) from the keyboard with their parameters, as follows:

```
TYPE EXAMPLE.CMD
COPY C:PROG1.TAB B:PROG2.TAB
TYPE B:PROG2.PRN
```

Notes:

1. If you want to specify more than 10 replaceable parameters, you may do so by using the SHIFT command, which is explained on page 3-28.
2. If you want to use % as part of a filename *within* a batch file, specify it twice. For example, to specify the file ABC%.EXE enter it as ABC%%.EXE in the batch file.

Defining Replaceable Parameters with Names

By using the SET command in a batch file, you can define replaceable parameters with "strings" or names instead of numbers. When you use a string instead of a number, you must begin and end the string with a percent sign in the places where you want the substitution performed. However, do not use percent signs when you define its value with SET. For example, suppose you have a batch file that has multiple occurrences of a filename. You would like to use this batch file at different times and specify a different filename each time. You can specify the filename each place it occurs in the batch file as:

```
%MYFILE%
```

Set the value for %MYFILE% with a SET command that precedes the occurrences of %MYFILE% in the batch file, for example:

```
SET MYFILE=MYFILE1.TXT
```

Then run your batch file. The batch processor substitutes each occurrence of this parameter with the value you defined with the SET command. When you want to specify a different filename, you need only change the SET command, for example:

```
SET MYFILE=MYFILE2.TXT
```

See page 2-117 for an explanation of the SET command.

You can also specify system variables as replaceable parameters in a batch file. For example, suppose you have this replaceable parameter in your batch file:

```
%PATH%
```

When the batch processor encounters this parameter, it replaces it with the current value in the environment for PATH.

Suppressing Echoing of Batch File Statements

The ECHO command allows you to suppress the display or "echoing" of batch file statements processing in either mode. (See page 3-13 for an explanation of the ECHO command.) You can also suppress echoing by typing an @ symbol as the first character of a batch file command. For example, if you type:

```
@CMDNAME
```

this causes the echoing of the line the command is on to be suppressed. If you enter:

```
@ECHO OFF
```

this suppresses the echoing of ECHO OFF, which in turn suppresses the lines that follow until an ECHO ON is encountered.

Another method to suppress echoing of batch file statements works only in OS/2 mode: typing a /Q in the syntax of the batch file command. For example, if you enter:

```
BATFILE.CMD /Q
```

this is equivalent to having ECHO OFF as the first statement of the batch file, except that no ECHO OFF message is displayed. An ECHO ON statement in the batch file causes echoing of batch file statements to resume. The /Q or "quiet parameter," can be placed anywhere in the argument list following the batch file name.

By including the @ symbol in the DO clause of the FOR statement, you can suppress the echoing of the iteration of the loop.

```
For %I IN (*.*) DO @ECHO %I
```

This example produces a simple list of files as output.

Batch Commands

The commands for batch files are listed alphabetically, starting with the next page.

CALL (nest batch file)

Purpose

Nests batch files.

Syntax



Parameters

batchfile The batch file to begin running.

arguments Information passed to the called batch file.

Remarks

Use the CALL command to nest batch files. The CALL command causes the currently running batch file's data structure and file pointer position to be saved, and a new data structure to be created. When the called batch file ends, the original batch file continues its processing with the statement following the CALL command.

Example

Here is an example of how the CALL command works. Suppose you have two batch files. Your first batch file is EXAMPLE1.CMD. It contains these commands:

```
ECHO EXAMPLE1.CMD is running...
CALL EXAMPLE2
ECHO EXAMPLE1.CMD is running again...
```

Your second batch file, the one you want to call, is EXAMPLE2.CMD. It contains:

```
ECHO EXAMPLE2.CMD is running now...
```

CALL (nest batch file)

If you are in the root directory when you enter EXAMPLE1 at the OS/2 command prompt, this is what you see on your screen:

```
[C:\]ECHO EXAMPLE1.CMD is running...
EXAMPLE1.CMD is running...

[C:\]CALL EXAMPLE2

[C:\]ECHO EXAMPLE2.CMD is running now...
EXAMPLE2.CMD is running now...

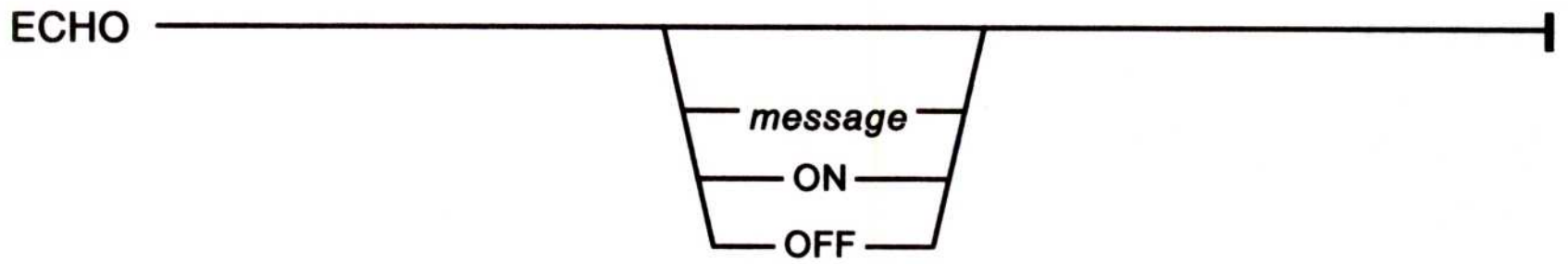
[C:\]ECHO EXAMPLE1.CMD is running again...
EXAMPLE1.CMD is running again...
```

ECHO

Purpose

Allows or prevents the screen display of OS/2 commands run from a batch file. It does not interfere with messages produced while commands are running.

Syntax



Parameters

message	Displays *message* regardless of the current ON or OFF state. In this way, you can display specific messages even when ECHO has been turned off.
ON	Displays all commands as they are run. This is the default value.
OFF	Stops the display of commands on the screen (including the REM command).

Remarks

Batch commands are normally displayed on the screen as they are read from the batch file. ECHO is ON after starting a batch file from the command prompt and is active only while the batch file is running.

If ECHO is issued with no parameters, the current ECHO state (ON or OFF) is displayed.

See "Suppressing Echoing of Batch File Statements" on page 3-10 for other methods you can use to prevent the display of commands during a batch file's processing.

ECHO

Example

In this example, the batch file contains the following:

```
ECHO OFF
REM **** command display is off
DIR C:/W
ECHO ON
DIR C:/W
```

When this batch file is run, the following display occurs:

```
ECHO OFF

The volume label in drive C is MYDISK.
 Directory of  C:\

FILE3       A     1613     7-17-87  12:14p
     1 Files    141312 bytes free

DIR C:/W

The volume label in drive C is MYDISK.
 Directory of  C:\

FILE3       A     1613     7-17-87  12:14p
     1 Files    141312 bytes free
```

In the above example, the ECHO OFF is displayed. The REM command and the first DIR C:/W are not displayed because ECHO is OFF, but the output of the DIR is still displayed. Then, ECHO ON is run, and the command DIR C:/W is displayed.

Related Commands

REM, PAUSE

ENDLOCAL (environment)

Purpose

Restores the drive, directory, and environment variables that were in effect before the SETLOCAL command was executed.

Syntax



Remarks

This command ends the SETLOCAL command, restoring the previous drive, directory, and environment settings. Issuing an ENDLOCAL command without a SETLOCAL command executing is not considered an error.

Example

The following example shows how the SETLOCAL and ENDLOCAL batch commands work together. LOCAL1.CMD is a batch file that contains these commands:

```
PATH
SETLOCAL
A:
CD \XYZ
PATH A:\;A:\MISCPGM
PATH
ENDLOCAL
PATH
```

ENDLOCAL (environment)

Suppose you are in the root directory of drive C and your current path is:

```
PATH=C:\;C:\OS2;C:\OS2\INSTALL
```

When LOCAL1.CMD is run, this is what is displayed at the OS/2 command prompt:

```
[C:\]PATH
PATH=C:\;C:\OS2;C:\OS2\INSTALL

[C:\]SETLOCAL

[C:\]A:

[A:\]CD \XYZ

[A:\XYZ]PATH A:\;A:\MISCPGM

[A:\XYZ]PATH
PATH=A:\;A:\MISCPGM

[A:\XYZ]ENDLOCAL

[C:\]PATH
PATH=C:\;C:\OS2;C:\OS2\INSTALL
```

The PATH command with no parameters displays the current value for PATH. You can see that the original current drive, directory, and environment value for PATH are saved when SETLOCAL is issued and restored when ENDLOCAL is issued.

Related Commands

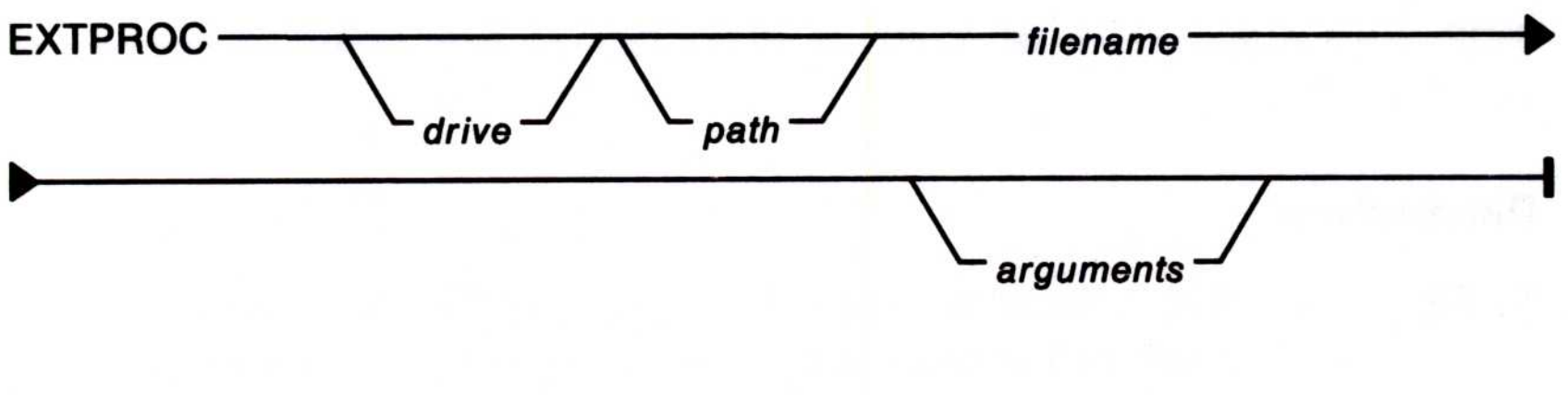
SETLOCAL

EXTPROC (external processor)

Purpose

Defines an external batch processor for a batch file.

Syntax



Parameters

filename	The complete name of the file, which includes its extension, that contains the external batch processor.
arguments	Information passed to the external batch processor.

Remarks

This command is useful if you have your own batch processor and wish to substitute it for the OS/2 batch processor. EXTPROC must be the first line of your batch file so that CMD.EXE can call it to process your batch file statements.

Example

Suppose you have written your own batch processor called MYBATCH.EXE, which is located in the C:\BATCH directory. You would like to use it to run some of your batch files. In each of the OS/2 mode batch files you want your batch processor to process, you must include this command as the *first statement* in the file:

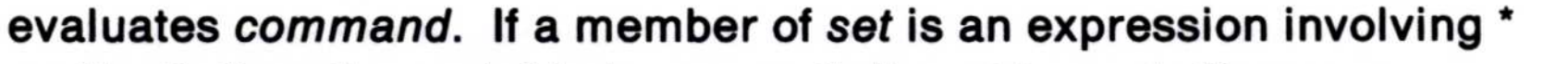
```
EXTPROC C:\BATCH\MYBATCH.EXE
```

FOR

Purpose

Allows repetitive execution of OS/2 commands.

Syntax



Parameters

%%c	Any character, including 0 through 9. Note that this does not conflict with the %0 through %9 batch parameter substitution since two % symbols (%%) are required with batch processing when doing the FOR loop substitution.
set	One or more items.
command	OS/2 command to be run.

Remarks

FOR sequentially sets the *%%c* variable to each member of *set*, then evaluates *command*. If a member of *set* is an expression involving * and/or ?, then the variable is sequentially set to each filename matching the pattern. In this case, only one item may be in the set, and any other items are ignored. %% is required if the FOR command is included in a batch file. To type the FOR command at the OS/2 prompt, only include one %. For example, you can type:

```
FOR %h IN (FILE1) DO DIR %H
```

Example

Suppose you type the following command in your batch file:

```
FOR %%f IN (PROG1.ASM PROG2.ASM PROG3.ASM) DO DIR %%f
```

When you run your batch file, the result is the same as the result of entering these three commands:

```
DIR PROG1.ASM
DIR PROG2.ASM
DIR PROG3.ASM
```

GOTO

Purpose

Transfers control to the line following the one containing the appropriate label.

Syntax



Parameters

label	Causes commands to be executed beginning with the line immediately after *:label*.

Remarks

A *label* is inserted in a batch file as a colon (:) followed by the label name. If *:label* is not defined, the current batch file ends. A *label* in a batch file is defined as a character string where the first eight characters are significant (make it different). A *label* cannot contain a period (.).

Example

In this example, the following batch file produces an indefinite sequence of REM **looping...** and GOTO **loop** messages on the screen:

```
:loop
REM looping...
GOTO loop
```

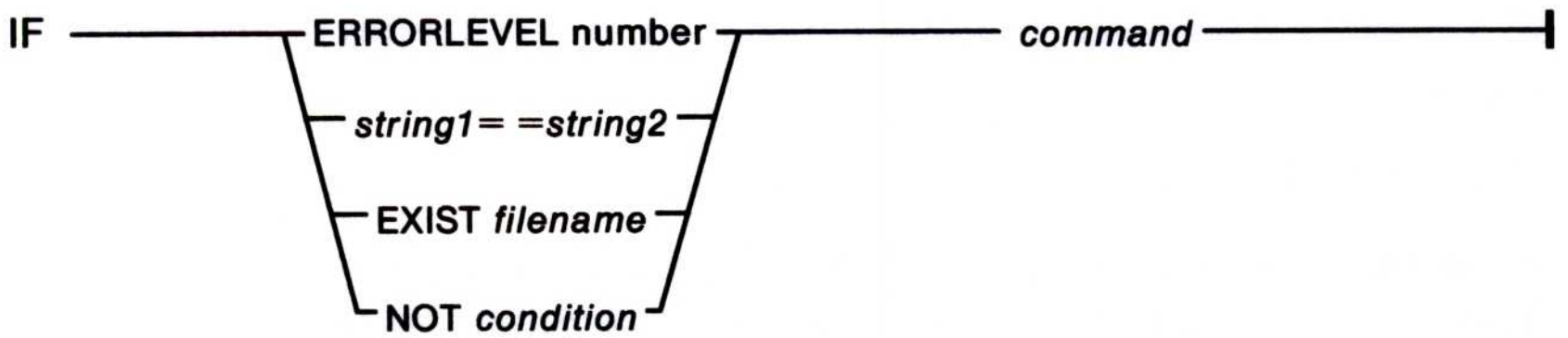
Related Commands

IF

Purpose

Allows conditional execution of OS/2 commands.

Syntax



Parameters

ERRORLEVEL *number*	True if the previous program had an exit code of *number* or higher. The number is specified as a decimal value.
string1 = = *string2*	True when *string1* and *string2* are identical.
EXIST *filename*	True if *filename* is found in the specified directory. Global characters (? and *) are allowed in *filename*.
NOT *condition*	True if the *condition* is false.
command	The OS/2 command to be run when one of the above conditional parameters is true.

Remarks

When the conditional parameter specified is true, the OS/2 command is run; otherwise, the OS/2 command is ignored.

IF ERRORLEVEL number: Suppose you have these two commands in a batch file:

```
MYPROG1
IF ERRORLEVEL 1 ECHO MYPROG1 FAILED
```

MYPROG1 is a program that sets the error level when it completes its processing. MYPROG1 sets the error level to 0 if processing is completed successfully; otherwise it sets the error level to 1. The batch

file IF ERRORLEVEL 1 ECHO ... tests for the condition where MYPROG1 fails. If MYPROG1 does not complete processing successfully, the condition is true and the ECHO batch command is executed. The ECHO batch command displays the data (or message) immediately following *ECHO*. If MYPROG1 is successful, the condition is not true, and the ECHO batch command is not executed. Processing would then continue with the next command in the batch file.

IF string1 = = string2: Suppose you have the following statement in a batch file:

```
IF %1 == JOHN ECHO John was here!
```

If the replaceable parameter **%1** is equal to JOHN, the condition is true. The ECHO batch command is then executed, displaying **John was here!** If JOHN is not given as the **%1** parameter, the condition is false. The ECHO batch command is not executed. Processing continues with the next command in the batch file.

IF EXIST filename: Suppose you have this statement in your batch file:

```
IF EXIST FILE1 GOTO ABC
       .
       .
       .
:ABC
command
```

The condition is true, provided FILE1 is found in the current directory on the default drive. The GOTO ABC is executed, causing the system to skip to the command following the label :ABC. If FILE1 is not found, the GOTO ABC is not executed. Processing then continues with the next command in the batch file.

IF NOT condition: Suppose you have the following in a batch file:

```
IF NOT EXIST A:%1 COPY B:%1 A:
PROG2
```

The program PROG2 requires a particular file to be on drive A. The IF command is executed prior to executing the PROG2 program to make sure that the required file is on drive A. If the file does not exist on drive A, the condition is true. The COPY command is then executed, copying the file from drive B to drive A to satisfy the requirements of the program. If the file does exist on drive A, the copy is not executed. Processing then continues with the execution of PROG2.

Related Commands

GOTO

PAUSE

Purpose

Suspends execution of the batch file and displays the message **Press any key when ready...**

Syntax



Parameters

comment The *comment* can be any string of characters up to 121 characters long.

Remarks

You can insert PAUSE commands within a batch file to display messages and to give you the opportunity to change diskettes between commands or perform some other action. To resume processing of the batch file, press any key.

You can control how much of a batch file you want to process by placing PAUSE commands at strategic points in the file. At each PAUSE command, the system stops and gives you time to decide whether to stop the batch from running.

You can stop a batch file from running in either OS/2 or DOS mode by pressing Ctrl+Break. However in an OS/2 session, you are not given the option of continuing the batch file's processing. See "Stopping a Batch File's Processing" on page 2-21.

Example

You can use PAUSE in a batch file to stop it and display a message:

```
PAUSE Change diskette in drive A
```

This example lets you change diskettes between commands. The message is displayed before the **Press any key when ready...** prompt.

Related Commands

ECHO, REM

REM (remarks)

Purpose

Displays remarks from within a batch file.

Syntax



Parameters

comment The *comment* is displayed when the batch execution reaches the REM command. If ECHO is OFF, then the remarks are not displayed. The only separators allowed in the *comment* are space, tab, and comma.

Remarks

You can use REM commands without remarks for spacing within your batch file, for readability.

Examples

If the following REM command is issued in a batch file, this remark is displayed:

```
REM This is the daily checkout program
```

To prevent OS/2 from interpreting characters, enclose them in double quotes. Only the first line causes a directory listing.

```
REM May not want to ¦ DIR
REM May not want to "¦" DIR
```

Related Commands

ECHO, PAUSE

SETLOCAL (environment)

Purpose

Lets you begin defining the drive, directory, and environment variables that are local to the current batch file.

Syntax



Remarks

This command saves the current drive, directory, and environment variables, and lets you define local variables for the batch file. The previous drive, directory, and environment settings are restored when the ENDLOCAL command is encountered, or when the batch file ends.

If a matching ENDLOCAL command is not found, the saved elements are restored when the batch file issuing the SETLOCAL command ends. In this way, any environment variable and the current drive and directory can be altered without affecting the command processor. Issuing multiple SETLOCAL commands without matching ENDLOCAL commands is not considered an error.

Example

The example for SETLOCAL appears under the discussion of the ENDLOCAL command on page 3-15.

Related Commands

ENDLOCAL

SHIFT

Purpose

Allows the use of more than 10 replaceable parameters in batch file processing.

Syntax



Remarks

Usually, command files are limited to handling 10 parameters, %0 – %9. To allow access to more than 10 parameters, use SHIFT to change the command line parameters. All parameters are shifted one position to the left, with the %0 parameter being replaced by the %1 parameter.

There is no backward shift. Once SHIFT is executed, the %0 parameter that existed before the shift cannot be recovered.

Example

Assume:

```
%0 = 'big'
%1 = 'golden'
%2 = 'car'
%3 – %9 are empty
```

then a SHIFT will result in the following:

```
%0 = 'golden'
%1 = 'car'
%2 – %9 are empty
```

For more information on using replaceable parameters in batch files, see "Using Replaceable Parameters in a Batch File" on page 3-8.

Chapter 4. Statements for the CONFIG.SYS File

System configuration is determined by a file called CONFIG.SYS. Operating System/2™ (OS/2™) has a set of statements for use only in the CONFIG.SYS file. These statements appear in alphabetic order starting on page 4-5 of this chapter. For an overview of CONFIG.SYS statements and their functions, see "Statements for the CONFIG.SYS File" on page A-7.

The CONFIG.SYS File

A single CONFIG.SYS file configures the system for both DOS and OS/2 modes. When you start the system, OS/2 looks for CONFIG.SYS in the root directory of the startup drive and interprets the statements within the file. In most cases CONFIG.SYS statements have system default values, which are assigned at system startup when:

- No CONFIG.SYS file is found
- A statement is not in the CONFIG.SYS file
- A statement is specified incorrectly, thus ignored.

These system defaults are shown on the syntax explanation pages under the heading "Default."

The following statements may be specified more than once:

- DEVICE
- DEVINFO
- REM
- RUN
- TRACE.

All other statements may be specified only once. If any of these statements are specified more than once, the last correct specification of the statement is used.

Warning: You should use caution in configuring a system for only OS/2 mode. If you do not have an OS/2 editor and you decide to reconfigure your system for DOS, you will have to rename your CONFIG.SYS file and restart OS/2 so default CONFIG.SYS values can be assigned. (The system editor EDLIN runs only in DOS mode.)

The Installation Program

The installation program creates a CONFIG.SYS file for you when it installs OS/2. To view the contents of this file, you can enter:

```
TYPE CONFIG.SYS
```

During system installation you are shown the statements that are created for your CONFIG.SYS file. In some cases the values placed in these statements by system installation differ from and override the system default values. These differences are intended to provide the best performance by OS/2 for *most* operating environments. There may be some instances where you do not want these preset values and instead need to enter other values. The installation program allows you to do this. For example, you may want to enter special values for CONFIG.SYS statements that have been recommended by the books that came with your OS/2 application programs.

Changing CONFIG.SYS

If you decide you want to make changes to your CONFIG.SYS file after system installation, you can use an editor to add, remove, or change statements. The system editor EDLIN, which runs in DOS mode, is described in Appendix C, "The Line Editor (EDLIN)" on page C-1. When you make changes to the CONFIG.SYS file, these changes do not take effect until the next time you start the system.

You can also make changes to CONFIG.SYS by reinstalling OS/2 with the installation program. In most cases you probably do not want to repeat the entire system installation process to make a few changes in CONFIG.SYS. However, you may consider rerunning the installation program if you have to change statements in CONFIG.SYS that are interrelated or device dependent. For example, suppose code page switching was not enabled during system installation and now you want to be able to use this feature of OS/2. By reinstalling OS/2, you

can let the installation program set up the interrelated DEVINFO, COUNTRY, and CODEPAGE statements in CONFIG.SYS for you. It uses the information provided by your responses to construct the appropriate statements for your CONFIG.SYS file.

Note: If you have decided to change your CONFIG.SYS file by reinstalling the system, refer to "Installing the System" in the *User's Guide* for this procedure. You cannot change every CONFIG.SYS statement by reinstalling the system; for example, you cannot change PROTSHELL or SHELL. The installation program always gives you the system defaults for these statements. To change them, you must use an editor.

Code Page Switching

To successfully switch between two code pages, you must ensure that the following statements are set up correctly in CONFIG.SYS.

CODEPAGE: This statement identifies the two code pages to be used for code page switching. These code pages must be the ones supported for the country specified by the country code in your COUNTRY statement. The table on page 4-11 lists countries and their supported code pages.

For instructions on how to enter the CODEPAGE statement, see the syntax explanation on page 4-8.

COUNTRY: The COUNTRY statement specifies the country code and the complete name of the file that contains a set of country information for each code page supported for that country. If no file is specified, the system defaults to the COUNTRY.SYS file in the root directory of the startup drive.

For instructions on how to enter the COUNTRY statement, see page 4-10.

DEVINFO: Separate DEVINFO statements are required for each device to be used for code page switching:

Keyboards: The statement for the keyboard specifies your keyboard layout and the file that contains a keyboard layout table for translating keystrokes into the characters of each code page supported by OS/2. The name of this file is KEYBOARD.DCP. There is no default filename.

Displays: The statement for the display specifies the display name and the file that contains a video font table for displaying characters in each of the code pages supported by OS/2. The name of this file is VIOTBL.DCP. There is no default filename.

Printers: The statement for the printer specifies the printer name and the file that contains a printer font table for each code page supported by OS/2. The name of this file is the printer number with DCP as its extension. There are no default filenames.: Not all devices are suitable for code page switching. For a list of the allowable devices and instructions on how to enter DEVINFO statements, see page 4-30.

Incorrect, partial, or mismatched setup of statements for code page selections, country code, keyboard layout, display, and printer may cause ineffective switching between code pages.

Statements for CONFIG.SYS

The statements for the CONFIG.SYS file are listed alphabetically, starting with the next page.

In most cases, the "Default" headings show the system default values assigned to a statement when:

- No CONFIG.SYS file exists
- A statement is omitted from CONFIG.SYS
- A statement is specified incorrectly.

There are some statements whose defaults are dependent on other statements in CONFIG.SYS, the type of computer you have, or the amount of storage installed in your computer. In these cases, this information is noted.

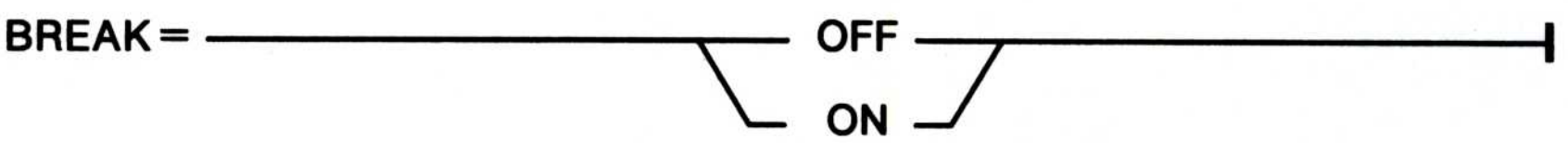
If a statement has a value assigned by the installation program that is intended to override the system default, this value is noted beneath the system default.

BREAK

Purpose

Instructs the system to check for Ctrl+Break at all times in DOS mode.

Syntax



Parameters

ON Enables the check for Ctrl+Break.

OFF Disables the check for Ctrl+Break. This means that OS/2 checks for Ctrl+Break being entered only during:

- Standard output operations
- Standard input operations
- Standard print operations
- Standard auxiliary operations.

Remarks

If you want the system to check for Ctrl+Break in DOS mode whenever it is requested during a batch file, set BREAK on. This allows you to stop a program that produces few or no standard device operations (such as a compiler). For instance, if a program is being compiled, it is important to have a way to stop compilation if an error or loop is encountered.

Note: In OS/2 mode, the check for Ctrl+Break is always set to on and you cannot change it.

BREAK



Default

BREAK = OFF

Example

To enable the check for Ctrl+Break in DOS mode, add this statement to CONFIG.SYS:

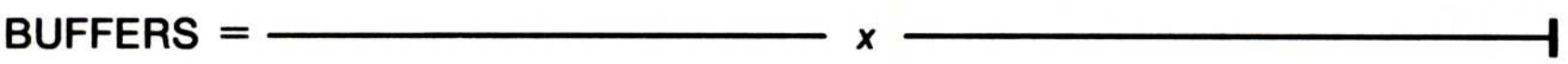
```
BREAK=ON
```

BUFFERS

Purpose

Determines the number of disk buffers that the system will use.

Syntax



Parameters

x	A number from 1 through 100. Values greater than 100 are ignored.

Remarks

The disk buffer is a 512-byte block of storage that OS/2 uses to read and write blocks of data that do not occupy an entire sector. You can increase the speed of your system by increasing the value specified for BUFFERS. However, when you increase the number of disk buffers, you decrease your available storage. This is a number you may want to experiment with to get maximum performance.

Default

BUFFERS=3

Note: If you accept the system installation default for the BUFFERS statement, the installation program places the following in your CONFIG.SYS file:

```
BUFFERS=30
```

Example

To change the number of buffers used by OS/2 to read and write data to and from disk to 10, change your BUFFERS statement to:

```
BUFFERS=10
```

CODEPAGE (prepare code pages)

Purpose

Selects the system code pages (defined character sets) to be prepared by OS/2 for code page switching.

Note: This statement is one of the interrelated CONFIG.SYS statements required for successful code page switching. For a list of these statements, see "Code Page Switching" on page 4-3.

Syntax



Parameters

xxx The primary code page.

yyy A secondary code page.

Remarks

Code pages supported by OS/2 are:

Code Page	
437	U.S. IBM PC
850	Multilingual
860	Portuguese
863	Canadian-French
865	Nordic

System code pages determine the characters entered by the keyboard, shown on the display, and printed on the printer. OS/2 supports keyboard, display, and printer code page switching in a multitasking environment.

If you want to see the characters contained in each of the code pages supported by OS/2, refer to Appendix F, "System Code Pages" on

page F-1. For an explanation of the purpose of code page switching, see the discussion under the CHCP command, "Why Switch Code Pages" on page 2-23.

Default

The display and printers each default to a native device character set. The keyboard and country information default to the national language code page supported by the country code included with the COUNTRY statement.

Example

Suppose you have a U.S. keyboard and a display and printer that support code page switching. The following statement in CONFIG.SYS would make both the U.S. IBM PC code page and the multilingual code page available to your system:

```
CODEPAGE=437,850
```

Of course, you need to include the appropriate DEVINFO statements in your CONFIG.SYS file to be able to use both code pages.

Related Commands

COUNTRY and DEVINFO statements in CONFIG.SYS, CHCP

Note: It is recommended that both files and subdirectories be named using only the characters A-Z and 0-9. This prevents file access problems when switching between code pages and their associated character capitalization rules.

COUNTRY

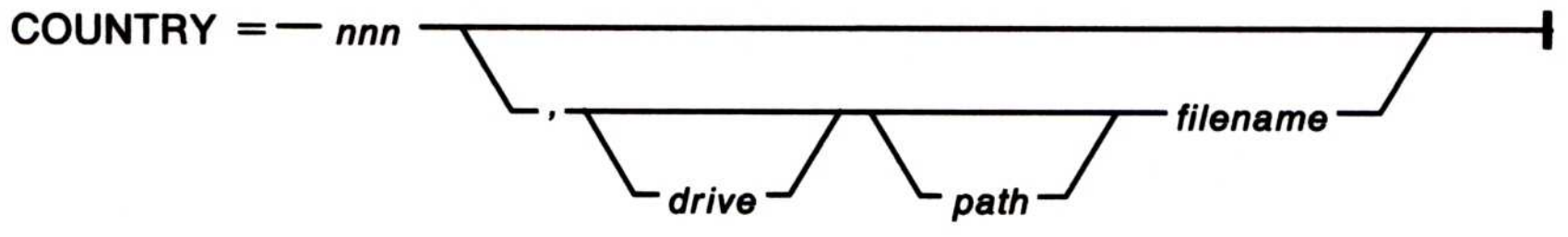
Purpose

Identifies the country for which country dependent information is selected:

- Date and time format
- Decimal separator
- Character case map table
- Collating sequence table used by SORT
- DBCS environment vector for double-byte characters.

Note: This statement is one of the interrelated CONFIG.SYS statements required for successful code page switching. For a list of these statements, see "Code Page Switching" on page 4-3.

Syntax



Parameters

nnn A three-digit value. For most countries, this value is the three-digit international country code for the telephone system.

filename The complete name of the file, which includes its extension, that contains country dependent information.

Remarks

If *filename* is not specified, country information is supplied by the default COUNTRY.SYS file in the root directory of the OS/2 startup drive. The default file contains information for each country that includes one set of country information for each system code page supported by that country.

Warning: If you specify a filename without a *path*, OS/2 looks for the specified country information file in the current directory of the default drive at the time it needs to look at the file. This may not be the root directory of the startup drive.

The country dependent information used by OS/2 is determined by the COUNTRY statement that is active in the system. This country information is always the same for both operating modes.

The table below lists the country code and supported code pages for each country. The first code page listed for each country is what the system defaults to when a CODEPAGE statement is not specified in CONFIG.SYS.

Country	Country Code	Code Pages Supported
Australia	061	437, 850
Belgium	032	437, 850
Canada (English)	001	437, 850
Canada (French)	002	863, 850
Denmark	045	865, 850
Finland	358	437, 850
France	033	437, 850
Germany	049	437, 850
Italy	039	437, 850
Latin America	003	437, 850
Netherlands	031	437, 850
Norway	047	865, 850
Portugal	351	860, 850
Spain	034	437, 850
Sweden	046	437, 850
Switzerland	041	437, 850
United Kingdom	044	437, 850
United States	001	437, 850

COUNTRY

Default

COUNTRY = 001

Note: This is the U.S. country code.

Example

To specify the date and time format for Canada (French), include the following command in the CONFIG.SYS file:

```
COUNTRY=002
```

The next time you start OS/2, the date format is *yy-mm-dd*, the time format is *hh:mm:ss*, and the decimal separator is a comma (,).

Related Commands

DATE, DIR, KEYB, SORT, TIME

DEVICE

Purpose

Specifies the path and filename of a device driver to be installed.

Syntax



Parameters

filename The complete name of the file, which includes its extension, that contains the device driver.

arguments Information to be passed to the device driver.

Remarks

The standard default device drivers loaded by OS/2 support the keyboard, display, printer, diskette, fixed disk, and clock devices. These device drivers must *not* be listed in CONFIG.SYS with DEVICE statements. To install other device drivers, DEVICE statements are required.

Both DOS mode device drivers and OS/2 mode device drivers may be loaded with the DEVICE statement. OS/2 device drivers are initialized when you start Operating System/2 and can process requests from either DOS or OS/2 programs. DOS device drivers are also initialized when you start Operating System/2 but can only process requests from DOS programs.

DEVICE statements are processed in the order in which they appear in CONFIG.SYS.

DEVICE

Device Installation Warning Messages: There can be several reasons why devices fail to install properly during system initialization. Warning messages are displayed to help you identify the problem. First, you get a message in this format:

```
< device > did not install.
```

where "device" is the name of the device that failed to initialize; for example, COM2. This message may indicate a normal condition when your system unit has only one asynchronous device available or another device driver has claimed a port for its use. However, this message can also be generated to indicate a problem with an aynchronous device that is installed. Additional messages are displayed to help you determine whether or not a problem exists.

- For example, suppose you attempted to load the wrong asynchronous communications device driver for your computer. Assume you have an IBM Personal System/2 and in your CONFIG.SYS file you have this statement:

  ```
  DEVICE=COM01.SYS
  ```

  This is the asynchronous communications device driver for the IBM Personal Computer AT and IBM Personal Computer XT Model 286. At system initialization you first receive a message telling you the device didn't install, followed by this message:

  ```
  The system unit is not an AT or XT-286.
  ```

  This means you must change your statement for the asynchronous communications device driver from COM01.SYS to the one required for the Personal System/2:

  ```
  DEVICE=COM02.SYS
  ```

  Then restart your system. If your computer had been an AT or an XT Model 286 and you made the similar mistake of loading the COMO2.SYS device driver, you would have received this message:

  ```
  The system unit is not a PS/2 machine.
  ```

- It is possible that you may have a faulty or incompatible asynchronous communications adapter installed that cannot be supported by the COMOX.SYS device driver. If this is the case, a message is displayed:

  ```
  The device adapter is not supported.
  ```

Consult with your service representative to correct the problem.

- If you get this message:

  ```
  The device adapter is not available.
  ```

  this means there is no asynchronous device available to support the COM port or the device is being used by another device driver.

  It is possible that this message indicates a normal condition, since the COM0XSYS device driver attempts to support ports COM1 through COM3 and you may be using one of these ports for another device driver. (See the example for the MOUSEB05.SYS device driver on page 4-26.) However, it is also possible that this message indicates a faulty or incompatible device driver. If the latter is the case, consult your IBM service representative to correct the problem.

- You may have a device driver that claims the same interrupt level required by the COM0XSYS device driver. When this is the case, a message is displayed:

  ```
  The interrupt level is not available.
  ```

  To correct the problem, you can remove the DEVICE statement for the device driver, or you can place its DEVICE statement *after* the one for COM0XSYS. It is possible that the device driver may not be compatible with COM0XSYS.

- If you receive this message:

  ```
  The Advanced BIOS is not present.
  ```

  It means the COM02.SYS device driver for the IBM Personal System/2 cannot access the ABIOS. If this condition occurs on machines that support ABIOS, consult with your IBM service representative to correct the problem. On machines that do not support ABIOS, the COM01.SYS device driver may be used.

DEVICE

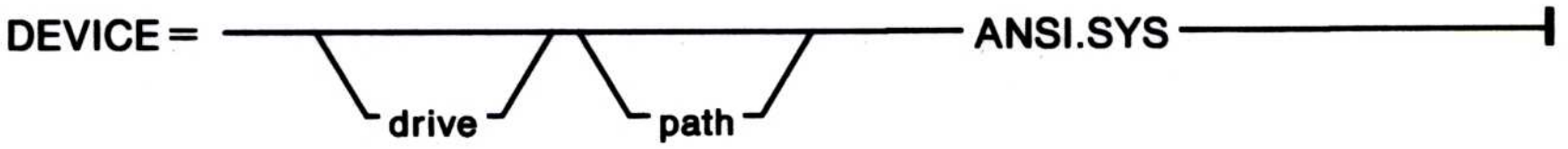
ANSI.SYS Device Driver

Purpose

Allows you to use extended keyboard and display support in DOS mode.

Note: This device driver has no effect on OS/2 mode.

Syntax



Remarks

If this DEVICE statement is in your CONFIG.SYS file when you start Operating System/2, the standard input and standard output support is replaced with the extended functions in DOS mode. These functions allow you to redefine keys, manipulate the cursor, and change display color attributes. If you decide you no longer want the extended functions for DOS, you can remove the DEVICE=ANSI.SYS from CONFIG.SYS and restart the system.

Note: For an explanation of how extended standard input and standard output is provided in OS/2 mode, see the ANSI command on page 2-7.

Related Commands

ANSI, PROMPT

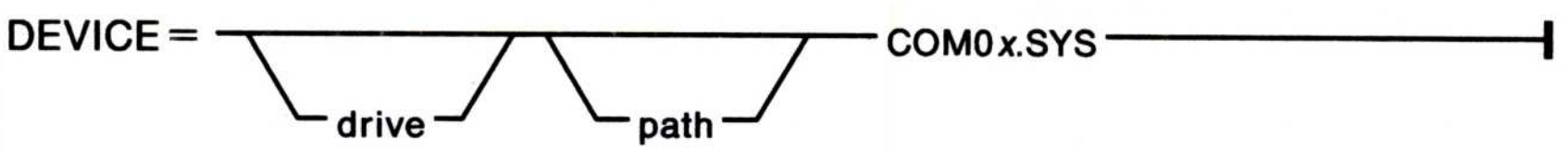
COM0x.SYS Device Driver

Purpose

Allows OS/2 application programs or system programs such as SPOOL to use serial devices.

Warning: If a device driver that supports any one of serial ports COM1 through COM3 is installed by a DEVICE statement that precedes a DEVICE=COM0X.SYS statement, the port will then be unavailable to COM0X.SYS.

Syntax



Parameters

x For the IBM Personal Computer AT and the IBM Personal Computer XT Model 286, this value must be 1. For the IBM Personal System/2, Models 50, 60, and 80, this value must be 2.

Note: The COM01.SYS system file is the device driver to use for the IBM Personal Computer AT and IBM Personal Computer XT Model 286. This device driver supports ports COM1 and COM2. The COM02.SYS system file is the device driver to use for the IBM Personal System/2. This device driver supports ports COM1, COM2, and COM3.

Remarks

This device driver allows an OS/2 application program to support full duplex communication. The device driver services the COM port in an interrupt driven manner and provides:

- Transmission and reception queues
- Automatic control modes for modem control signals
- Logical data stream flow control (XON/XOFF) for transmit and receive.

DEVICE

COM0X.SYS does not provide any support for devices that are attached to the COM port. Instead, it provides enabling support for the asynchronous communications interface itself. OS/2 application programs, subsystems, and system programs provide the support needed to use devices attached to the COM port.

This device driver has a feature that allows additional device drivers to be installed that support specific COM ports. This feature is included to allow support for:

- A special device for an application program's use that is not adequately supported by this device driver.
- Additional COM ports besides COM1 through COM3.
- Enhanced device driver function for a COM port. (This may be required for certain subsystem support.)

Warning: This device driver uses storage that can also be used for DOS mode. Do not use this device driver unless you require its function, since it reduces the amount of storage available to run a DOS application.

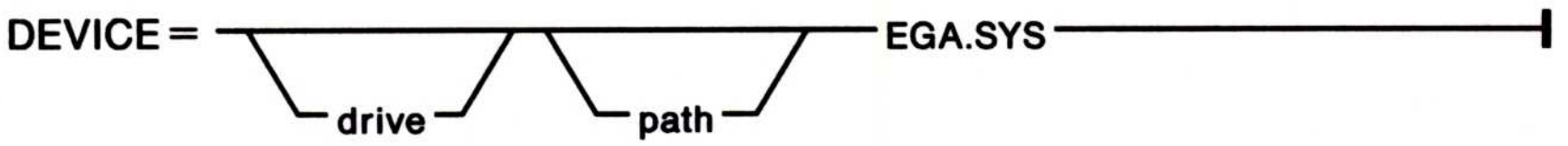
Device Installation Warning Messages: If a device driver fails to install properly, warning messages are generated during system initialization. Refer to "Device Installation Warning Messages" on page 4-14 for an explanation of these messages.

EGA.SYS Device Driver

Purpose

Supports the EGA register interface.

Syntax



Remarks

The EGA.SYS device driver provides support for the EGA register interface in DOS mode. EGA.SYS must be installed for those application programs that use the EGA register interface.

Note: See "Using Applications With Enhanced Graphics Features" on page B-8 for information on DOS mode application programs that use the Enhanced Graphics Adapter.

Related Commands

DEVICE = MOUSE*xxx*.SYS and DEVICE = POINTDD.SYS statements in CONFIG.SYS

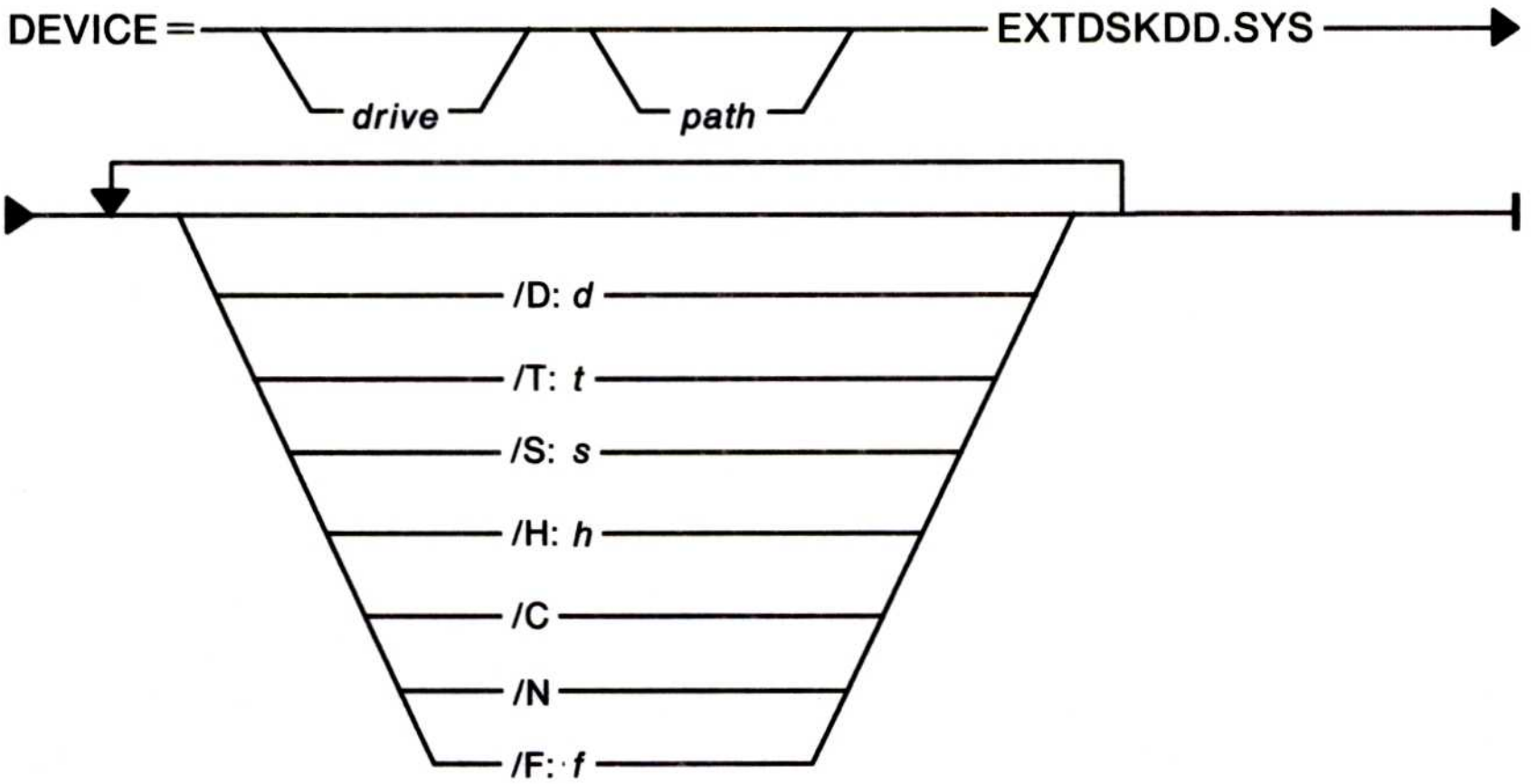
DEVICE

EXTDSKDD.SYS Device Driver

Purpose

Allows you to access a disk using a logical drive letter.

Syntax



Parameters

/D:*d* Specifies the physical drive number, which can have a value of 0 through 255.

To specify the first physical diskette drive, accessed as A from the command line, specify the value 0. The value 1 specifies the second physical diskette drive, and the value 2 specifies the third physical diskette drive, which must be external.

To specify the first physical fixed drive, use the value 128. The value 129 specifies the second physical fixed drive.

/T:*t* Is the number of tracks per side. The valid range is from 1 through 999. The default is 80.

/S:*s* Specifies the number of sectors per track. The valid range is from 1 through 99. The default is 9.

/H:*h* Is the maximum number of heads. The valid range is 1 through 99. The default is 2.

/C Indicates that changeline support is required. This is meaningful only on machines that support diskette changeline, such as the IBM Personal Computer AT

/N Specifies that the physical device is a nonremovable block device, such as a fixed disk.

/F:*f* Is the device type (form factor). Choose from the list below. The default is 2.

Value	Device
0	160/180Kb
0	320/360Kb
1	1.2Mb
2	720Kb

Remarks

More than one external device driver can be installed at the same time. Installed virtual disks will not affect the drive letter assignments described below if DEVICE=VDISK.SYS commands are listed *after* DEVICE=EXTDSKDD.SYS commands in the CONFIG.SYS file.

Example

The following examples apply to the IBM Personal Computer AT.

To set up logical drive D for a 720Kb external diskette drive when you have one internal diskette drive and one fixed disk, enter the following command in the CONFIG.SYS file:

```
DEVICE=EXTDSKDD.SYS /D:2
```

If, in the above example, you want to be able to copy from the external diskette drive to the same drive, enter the above command in the CONFIG.SYS file twice:

```
DEVICE=EXTDSKDD.SYS /D:2
DEVICE=EXTDSKDD.SYS /D:2
```

The logical drive letters D and E are assigned to the external diskette drive.

You can also use EXTDSKDD.SYS to copy from an internal diskette drive to the same drive. Assume you have a 1.2Mb drive as the first diskette drive, a 320/360Kb drive as the second diskette drive, and a fixed disk. Enter:

```
DEVICE=EXTDSKDD.SYS /D:0 /T:80 /S:15 /H:2 /C /F:1
```

This assigns the logical drive letter D to the first diskette drive, which can now be referenced as A and D. If from the default (A) command line you enter:

```
COPY FILE1 D:
```

You are prompted to insert the diskette for the appropriate logical drive, and FILE1 is copied from one diskette to another using only the 1.2Mb drive.

Rules for Drive Letters Assigned to Device Drivers: The first physical internal diskette drive is assigned A. The second internal diskette drive is assigned B. Letters from C on are assigned in the order devices or device drivers are encountered. The existence of internal diskette drives and fixed disks is checked first; then the CONFIG.SYS file is checked for device drivers. For OS/2 to recognize an external physical device, the CONFIG.SYS file must have the correct device driver information.

The drive letter B is automatically used, even if there is only one physical drive; thus, a machine with only one diskette drive has two logical diskette drives A and B. In this case, a specified parameter of /D:1 is an error. The first fixed disk or the first block device driver cannot have a drive letter lower than C assigned.

For machines with an external drive, if the external device driver is loaded twice, where D:*d* is the same, this creates two logical drives for the one physical drive, giving you the ability to transfer data from one diskette to another using the same drive.

This same concept can be applied to internal drives. OS/2 automatically loads a diskette drive device driver for the drive at setup time. If you include a DEVICE=EXTDSKDD.SYS in the CONFIG.SYS file for the same drive, two drive letters are associated with the same drive. The command

```
DEVICE=EXTDSKDD.SYS /D:0
```

in the CONFIG.SYS file at startup time causes OS/2 to load another device driver for the first diskette drive. The drive letter depends on the number of diskette drives and fixed disks in the machine. For a machine with two diskette drives and a fixed disk, the logical drive letter for the first diskette drive is D. With this setup you can copy files from the first physical diskette drive to the first logical diskette drive by referring to it as A and D.

DEVICE

MOUSExxx.SYS Device Driver

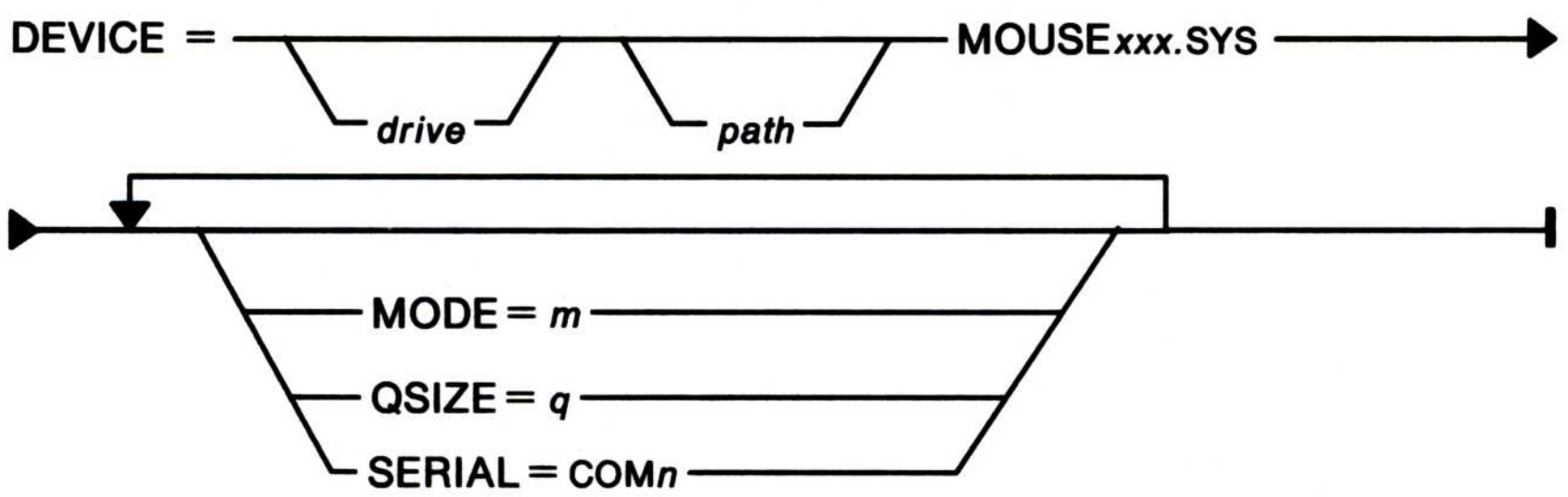
Purpose

Implements support for pointing devices.

Note: To effectively use a mouse, you must also load the POINTDD.SYS device driver.

Warning: DEVICE=MOUSE*xxx*.SYS statements for serial mouse device drivers must *precede* any COM0*x*.SYS statements in the CONFIG.SYS file, so that both device drivers will install correctly.

Syntax



Parameters

MODE = *m* Informs the mouse device driver whether mouse support is for OS/2 mode, DOS mode, or both. The default value is for both. The values indicating the type of support are:

- P = OS/2 mode only
- R = DOS mode only
- B = both operating modes.

QSIZE = *q* Specifies the length of the queue for events to be used for all OS/2 mode tasks. The valid range is 1 through 100. The default value is 10 event records.

SERIAL=COM*n* Specifies the serial communications port that the serial mouse is connected to. The valid values are COM1 and COM2 for the IBM Personal Computer AT, and COM1 through COM8 for the IBM Personal System/2. The default value is COM1 for a serial mouse. This parameter is not valid for a parallel mouse (Microsoft Bus Mouse, for example). A COM port connected to a mouse cannot then be supported by the COM0X.SYS device driver.

Remarks

Each of the supported devices is supplied with a device driver containing function for both DOS and OS/2 modes.

The mouse device drivers for the IBM Personal Computer AT and the IBM Personal Computer XT Model 286 are:

MOUSEA00.SYS	PC Mouse[1] by Mouse Systems Serial Mouse (part number 900120-214)
MOUSEA01.SYS	Visi-On[2] Mouse Serial Mouse (part number 69910-1011)
MOUSEA02.SYS	Microsoft[3] Mouse for IBM Personal Computers Serial Mouse (part numbers 039-099, 039-199)
MOUSEA03.SYS	Microsoft Mouse for IBM Personal Computers Parallel Mouse (part numbers 037-099, 037-199)
MOUSEA04.SYS	Microsoft Mouse for IBM Personal Computers In-Port Mouse (part number 037-299)

The mouse device drivers for the IBM Personal System/2, Models 50, 60, and 80 are:

MOUSEB00.SYS	PC Mouse by Mouse Systems Serial Mouse (part number 900120-214)
MOUSEB01.SYS	Visi-On Mouse Serial Mouse (part number 69910-1011)

---

1 PC Mouse is a trademark of Metagraphics/Mouse Systems.

2 Visi On is a trademark of VisiCorp.

3 Microsoft is a registered trademark of Microsoft Corporation.

DEVICE

MOUSEB02.SYS Microsoft Mouse for IBM Personal Computers Serial Mouse (part numbers 039-099, 039-199)

MOUSEB05.SYS IBM Personal System/2, Models 50, 60, and 80 In-board Mouse

Examples

Suppose you have an IBM XT Model 286, and you want to install MOUSEA00.SYS to operate in both OS/2 and DOS modes. Add this statement to your CONFIG.SYS file:

```
DEVICE=MOUSEA00.SYS MODE=B
```

Suppose you have an IBM Personal Computer AT, and you want to install MOUSEA01.SYS to operate in OS/2 mode only and use serial port 1. Add this statement:

```
DEVICE=MOUSEA01.SYS MODE=P SERIAL=COM1
```

Suppose you have an IBM Personal System/2, and you want to install MOUSEB05.SYS to work in both modes and use COM port 2. You also want to install the COM0X.SYS device driver. You add these statements in the order shown to your CONFIG.SYS file:

```
DEVICE=MOUSEB05.SYS MODE=B SERIAL=COM2
DEVICE=COM02.SYS
```

When your system is started, the mouse device driver is loaded and takes the COM2 port, leaving COM1 and COM3 ports available for the COM02.SYS device driver to use. You will then get a message as the COM02.SYS *device driver* installs, that tells you the COM2 *port* did not install. This message is followed by an additional message:

```
The device adapter is not available.
```

The reason the device adapter is not available is that the port has been claimed by the mouse.

For more information concerning these device messages displayed during system startup, see "Device Installation Warning Messages" on page 4-14.

Related Commands

DEVICE=COM0X.SYS statement in CONFIG.SYS

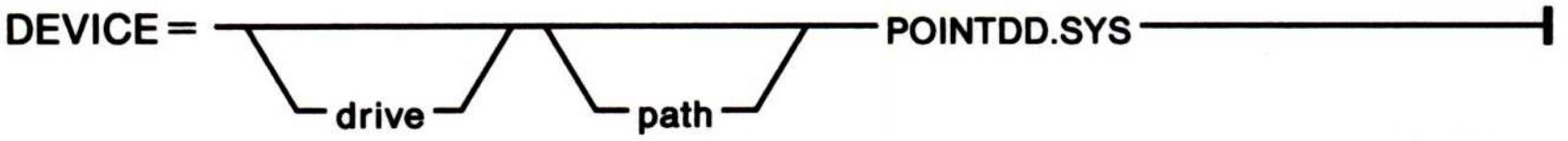
POINTDD.SYS Device Driver

Purpose

Provides mouse pointer draw support.

Note: To effectively use a mouse, in addition to loading MOUSE*XXX*SYS, you must also load this device driver.

Syntax



Remarks

POINTDD.SYS provides mouse pointer draw support. In OS/2 mode, text modes 0, 1, 2, 3, and 7 (and the + and * variations of these modes) are the only modes supported. In DOS mode, graphics modes 4, 5, 6, D, E, F, and 10 are also supported. For graphics modes D, E, F, and 10, the EGA.SYS device driver must also be installed, and the DOS application program to be used must support the EGA register interface.

Related Commands

DEVICE=MOUSE*XXX*.SYS and DEVICE=EGA.SYS statements in CONFIG.SYS

DEVICE

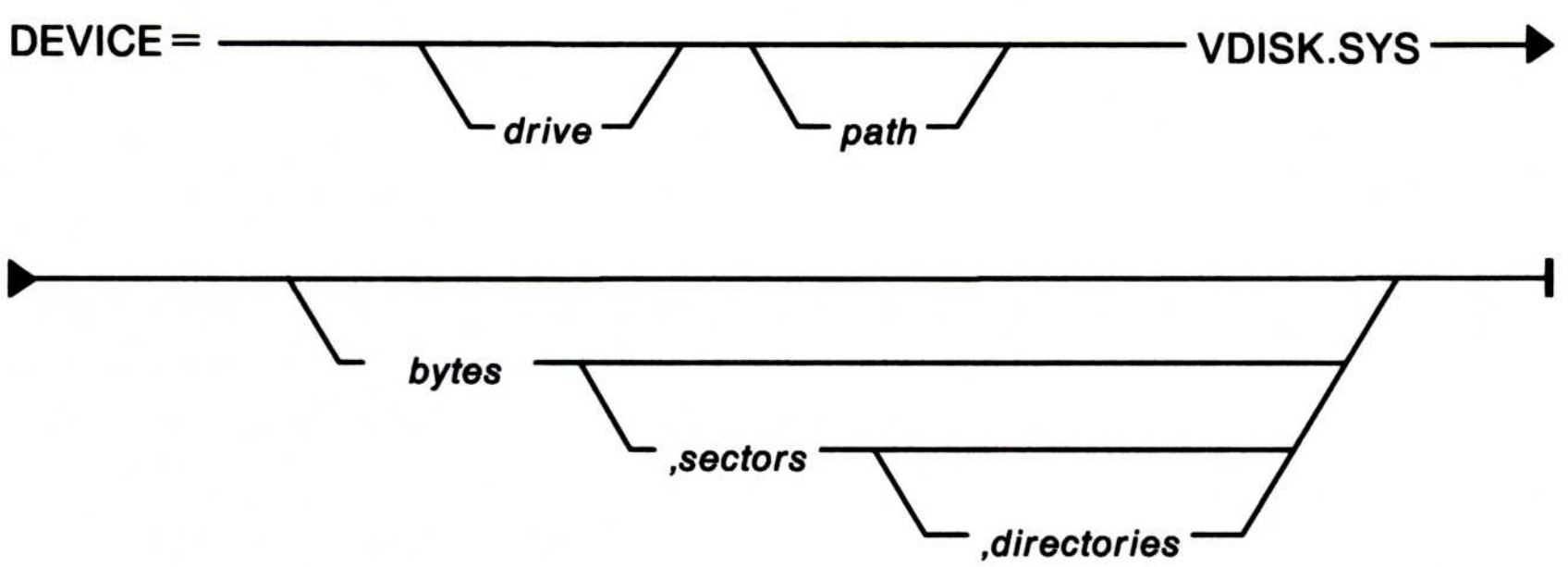
VDISK.SYS Device Driver

Purpose

Installs a virtual disk.

Warning: DEVICE=VDISK.SYS statements must be listed *after* any DEVICE=EXTDSKDD.SYS statements in CONFIG.SYS to avoid affecting logical drive letter assignments.

Syntax



Parameters

bytes Specifies the size of VDISK in K-bytes. The valid range is 16 through 4096. The default value is 64.

sectors Specifies the sector size in bytes. The allowed values are 128, 256, 512, and 1024. The default value is 128.

directories The number of directory entries. The valid range is 2 through 1024. The default value is 64.

Warning: The FAT file system cannot support a root directory containing more than 255 sectors. For example, a 64Kb disk with 128-byte sectors and 1024 directory entries requires 256 sectors. For 128-byte sectors, the maximum number of root directory entries you should specify is 1020.

Note: The parameters *bytes, sectors* and *directories* are positional parameters. This means that if a parameter is omitted but the one

that follows it is specified, the omitted parameter's position must be designated by a comma. See the example below.

Remarks

The value of *directories* is rounded up to the nearest sector size boundary. For example, if you give a value of 25, and your sector size is 512 bytes, 25 will be rounded up to 32, which is the next multiple of 16. (There are 16 32-byte directory entries in 512 bytes.)

If the virtual disk size specified is too large to fit in storage, VDISK will try to make a virtual disk 16Kb in size. This may result in a virtual disk with a different number of directory entries than was specified by *directories*.

Examples

The following example shows how to install a 160Kb virtual disk with 512-byte sectors and 64 directory entries.

```
DEVICE=VDISK.SYS 160,512,64
```

VDISK displays an informative message to tell you that it is installed. It also tells you the drive letter that is assigned to the virtual disk.

Suppose you want to specify a value for *directories* and accept the defaults for *bytes* and *sectors*, the value for *directories* should be preceded by two commas, as in this example:

```
DEVICE=VDISK.SYS ,,32
```

DEVINFO (code page device)

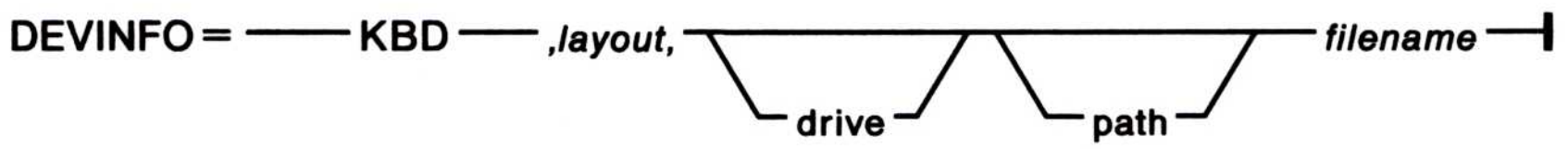
Purpose

Prepares a device for system code page switching.

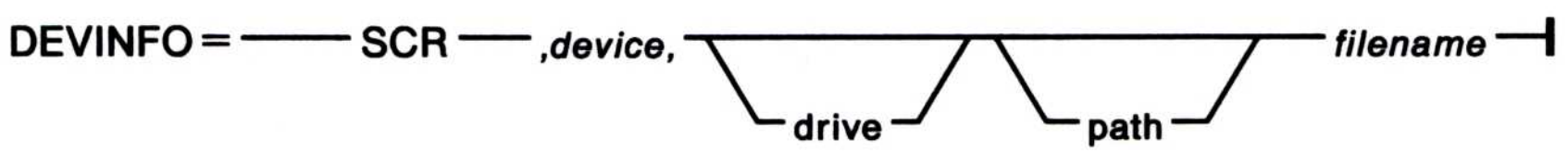
Note: This statement is one of the interrelated CONFIG.SYS statements required for successful code page switching. For a list of these statements, see "Code Page Switching" on page 4-3.

Syntax

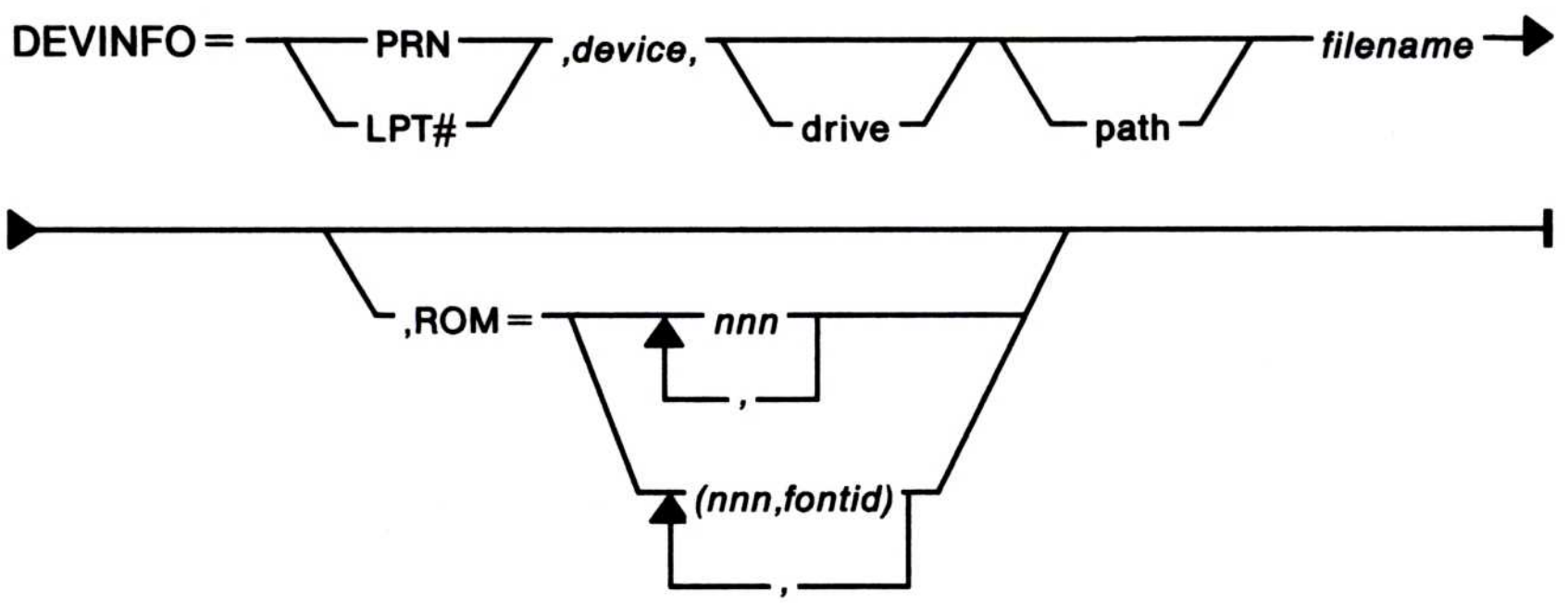
Keyboards:



Displays:



Parallel Printers:



Parameters

Keyboards:

layout Specifies the keyboard layout. See the KEYB command on page 2-81 for keyboard layout values.

filename Is the complete name of the file, which includes its extension, that contains the keyboard translation tables. The name of the system file is KEYBOARD.DCP. There is no default.

Displays:

device Is the reserved device name. The allowable device names are:

EGA IBM Enhanced Graphics Adapter
VGA IBM Personal System/2 Video Graphics Array
BGA IBM Personal System/2 Display Adapter

filename Is the complete name of the file, which includes its extension, that contains the system code pages for the display. The name of the system file is VIOTBL.DCP. There is no default.

Parallel Printers:

LPT# # is printer number 1, 2, or 3.

device Is the physical device number. The allowable device numbers are:

4201 IBM Proprinter™ Models 1 and II
5202 IBM Quietwriter® III

filename Is the complete name of the file, which includes its extension, that contains the system code pages for the printer. The name of the system file is the device number of the printer with DCP as its extension. For

---

Proprinter is a trademark of the International Business Machines Corporation.

Quietwriter is a registered trademark of the International Business Machines Corporation.

example, the filename to specify for the IBM Proprinter is 4201.DCP. There are no defaults.

ROM Specifies that system code pages are available in device read-only storage or cartridge.

nnn Represents a value for a system code page that is available in the printer device read-only storage. A printer device may support more than one system code page.

fontid Is the font identification number to be associated with the system code page. A printer may support more than one system code page. One system code page may have more than one font (a set of typographic characters all of one size and style) associated with it.

Note: For the IBM Quietwriter III, the *fontid* is a three-digit number on the label of the cartridge. The fontids for the imbedded code page 437 font are 011, 085, 159, and 254.

Remarks

If you do not correctly specify the required filenames or if you specify incorrect code pages in your CODEPAGE statement (those other than the two supported by the country code in your COUNTRY statement), then the device is not prepared for code page switching. If *drive* and *path* are not specified, OS/2 looks for the file in the root directory of the default drive.

Note that the keyword ROM is only meaningful for printers. Each ROM specification may be a pair of values enclosed in parentheses, where the first value is the system code page, and the second value is the number of a font that is in read-only storage or in a cartridge installed in the printer. If multiple fonts are not used, specify only the system code pages separated by a comma and without the parentheses.

Note: If there is a DEVINFO statement for your printer and the print spooler is active for the printer, all output on that printer will default to the printer's Character Set 2. Using Character Set 2 is necessary to support printing in either of the two OS/2 code pages specified with the CODEPAGE statement in CONFIG.SYS. Character Set 1 can still be

accessed using escape sequences. Refer to the instructions that came with your printer.

Examples

Suppose you have a U.S. keyboard, a VGA display, and a Proprinter Model II. You have already added CODEPAGE=437,850 to your CONFIG.SYS file to prepare the code pages. Now you want to prepare your keyboard, printer, and display to use them. Add the following statement to prepare your keyboard:

```
DEVINFO=KBD,US,C:\KEYBOARD.DCP
```

and this statement to prepare the display:

```
DEVINFO=SCR,VGA,C:\VIOTBL.DCP
```

and this statement to prepare your printer:

```
DEVINFO=LPT1,4201,C:\4201.DCP,ROM=(437,0)
```

Now restart OS/2, and you are ready to switch code pages.

If you want to use other fonts beside the defaults, the ROM specification for printers should include code page and font combinations in the correct order (imbedded fonts first), so that the application program will select the font you intended. For example, suppose you have an IBM Quietwriter III, and you wish to set it up to use code page 850 and 860, in addition to 437. Here is how you enter your DEVINFO statement:

```
DEVINFO=LPT1,5202,5202.DCP,ROM=(437,011),(437,085),(437,254),
  (437,159),(850,254),(850,159),(860,011),(860,085)
```

Note: For the purpose of readability, the above statement is shown as two lines. However, if it were an actual statement in your CONFIG.SYS file, it must appear as one line.

Related Commands

CODEPAGE and COUNTRY statements in CONFIG.SYS, CHCP, KEYB

DISKCACHE

Purpose

Specifies the number of blocks of storage to allocate for control information and use by the disk cache.

Note: DISKCACHE is supported only by IBM Personal System/2, Models 50, 60, and 80.

Syntax



Parameters

n	A number from 64 through 7200 that specifies the number of 1024-byte blocks of storage to be used for control information and use by the disk cache.

Remarks

If you have an IBM Personal System/2, this statement can be used to increase the speed of your system. However, when you increase the size of your disk cache, you decrease the size of available storage. You may want to experiment with the number of blocks you specify to get maximum performance. For a PS/2 with a total fixed disk size of 20Mb, a DISKCACHE of at least 64 is recommended. For total fixed disk size that is greater than 20Mb, a DISKCACHE of at least 128 is recommended.

The amount of storage required for control information is determined by the total fixed disk size of one or more fixed disks. If the amount of storage specified in *n* is not available, an error message is displayed. If the amount of storage specified in *n* is not sufficient to support the total fixed disk size, then diskcache will be disabled. The disk cache is allocated at system startup, and there is no dynamic adjustment of its size.

The disk cache allows a portion of the computer's system storage to be used as an additional fixed disk buffer. It speeds up application

programs that read fixed disks by keeping fixed disk data that is accessed frequently in a cache buffer. When an application program requests fixed disk data that is already in the cache buffer, the disk cache sends the data directly to the application program. This method of accessing data is much faster than if the data had to be read from the disk each time.

The IBM Personal System/2, Models 50, 60, and 80 disk device driver provides direct access storage disk cache support for use with fixed disks. This device driver controls operation of the cache, which is allocated by OS/2 and taken out of storage above 1Mb. The cache is run as a write-through cache to maintain data integrity: sectors passed to the disk device driver for writing on the disk are guaranteed to be written out to the disk upon successful return from the device driver.

Default

No disk cache is allocated.

Note: If you install on an IBM Personal System/2, Model 50, 60, or 80, and you accept the system installation default for DISKCACHE, the installation program places the following in your CONFIG.SYS file:

```
DISKCACHE=64
```

Example

You have an IBM Personal System/2 Model 60, and you want to set your disk cache size to 128Kb. Change your DISKCACHE statement to:

```
DISKCACHE=128
```

FCBS (file control blocks)

Purpose

Determines File Control Blocks (FCBS) management information for DOS mode.

Note: This statement has no effect on OS/2 mode.

Syntax

FCBS = ——— *m,n* ———

Parameters

m — A number from 1 through 255 that specifies the total number of file control blocks (FCBS) that can be open concurrently.

n — Specifies the number of files opened by FCBS that cannot be closed automatically by the system when a program tries to have more than *m* files opened by FCBS at one time. The first *n* files opened by FCBS are protected from being closed. The range of values for *n* is from 0 through 255. The value for *n* must be less than or equal to *m*.

Remarks

A file control block is a record that contains all of the information about a file; for example, its structure, length, and name. If a program tries to open more than *m* files, OS/2 closes the least recently used FCB and opens the new file. The first *n* files are not included in the list of least recently used FCBS. If a program tries to read or write to a file that has been closed because it is the least recently used FCB, then OS/2 displays an error message.

Note: New programs written for OS/2 should use internal file IDs for file input/output. For an explanation, see "Redirection in OS/2 Mode" on page 5-5.

FCBS (file control blocks)

Default

FCBS = 16,8

Example

If you decide you want to specify the number of file control blocks that can be open as 4, change your FCBS statement to:

```
FCBS=4,0
```

IOPL (input/output privilege)

Purpose

Allows I/O privilege to be granted to requesting processes in OS/2 mode.

Syntax



Parameters

NO	Prevents I/O privilege from being granted.
YES	Permits I/O privilege to be granted.

Remarks

The privilege level assigned to a program determines what code segments and data segments it can access. The privilege level also limits the machine instructions a program can execute.

Application programs are usually assigned privilege level 3. This means they can call routines that run at any other privilege level. However, they can reference only their own data segments and cannot issue any I/O instructions.

Programs that are granted I/O privilege run at privilege level 2. When IOPL=YES is specified, this permits a program that has been assigned privilege level 2, such as a subsystem that needs to communicate directly with a specific device, to send or receive instructions to or from that device.

Default

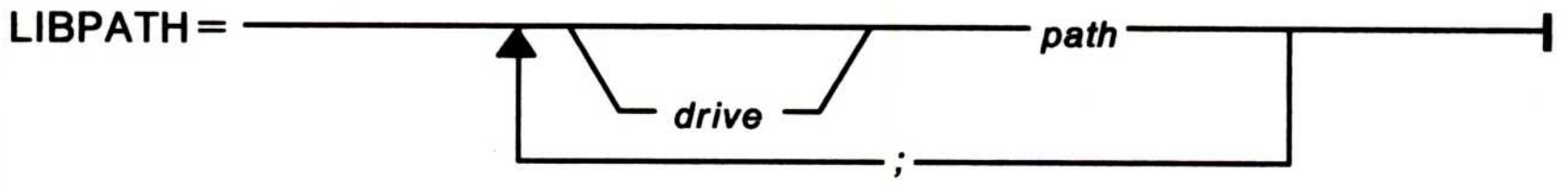
IOPL=NO

LIBPATH (library search path)

Purpose

Identifies the locations of dynamic link libraries for OS/2 programs.

Syntax



Parameters

drive path Specifies a directory to be searched for dynamic link libraries. More than one directory may be searched by specifying additional paths separated by semicolons. The paths are searched in the order specified.

Remarks

LIBPATH is used to identify a set of directories to be searched when loading dynamic link libraries. Since dynamic link library modules are shared globally, this operand allows path searching to be defined globally rather than on a per process basis as done by the PATH command.

Unlike the PATH environment variable, the current directory is not searched first.

Default

LIBPATH=C:\ if starting from a fixed disk
LIBPATH=A:\ if starting from a diskette

Note: The installation program places this statement in your CONFIG.SYS file:

```
LIBPATH=C:\;C:\OS2;C:\OS2\INSTALL;
```

LIBPATH (library search path)



Example

Suppose you have placed all your dynamic link libraries in a directory called \LIB. Change your LIBPATH statement in CONFIG.SYS so that OS/2 can find them:

```
LIBPATH=C:\; C:\LIB
```

MAXWAIT (maximum wait)

Purpose

Sets the time limit for lack of access to the processor resource in OS/2 mode.

Syntax



Parameters

x	An integer number in seconds that must elapse before a regular class thread may receive a boost in priority because of a lack of access to the processor. The valid range is 1 through 255.

Remarks

When a regular class thread is denied the processor for a defined number of seconds, it receives a temporary boost in priority for a minimum time slice.

Default

MAXWAIT=3

Example

The statement

```
MAXWAIT=10
```

causes a process to wait up to 10 seconds before it receives a temporary boost in priority.

Related Commands

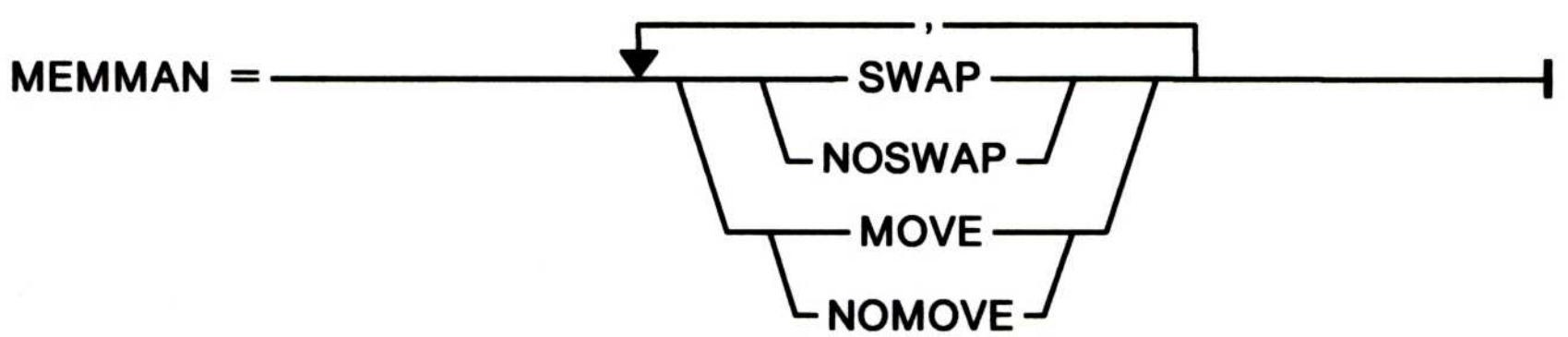
PRIORITY and TIMESLICE statements in CONFIG.SYS

MEMMAN (storage allocation)

Purpose

Selects storage allocation options for OS/2 mode.

Syntax



Parameters

SWAP	Permits segment swapping.
NOSWAP	Prevents segment swapping.
MOVE	Permits storage compaction.
NOMOVE	Prevents storage compaction.

Remarks

These parameters may be specified in any order. However, only one parameter from each pair (SWAP/NOSWAP, MOVE/NOMOVE may be specified. If you specify MEMMAN=SWAP, storage compaction is automatically permitted.

To ensure system responsiveness in those OS/2 environments where timing is critical (for example, process control tasks are being performed), storage compaction and/or swapping may be prevented.

Default

SWAP,MOVE if starting from a fixed disk
NOSWAP,MOVE if starting from a diskette



MEMMAN (storage allocation)

Example

Suppose you want to run a time dependent application. To prevent OS/2 from swapping the contents of storage to disk, add this statement to CONFIG.SYS

```
MEMMAN=NOSWAP
```

Related Commands

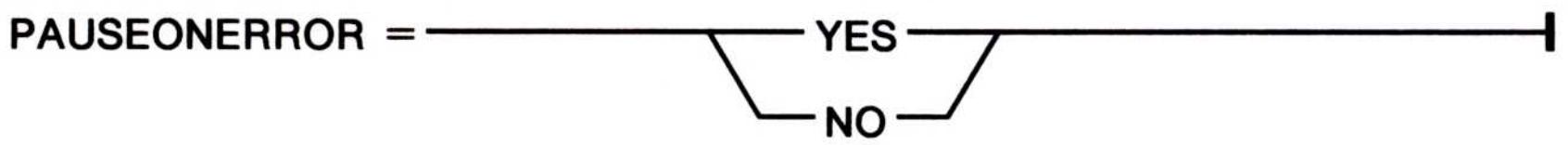
SWAPPATH statement in CONFIG.SYS

PAUSEONERROR (error prompting)

Purpose

Enables or disables pausing when error messages are issued during the processing of the CONFIG.SYS file.

Syntax



Parameters

YES If any errors have been detected in CONFIG.SYS, the system pauses just before starting the specified user interface or command processor and displays a prompt. Processing does not continue until you press Enter.

Note: The system does not pause when only warning messages are issued.

NO The system does not pause before starting the user interface or command processor.

Remarks

PAUSEONERROR=YES allows you to read any error messages generated during the processing of your CONFIG.SYS file.

Default

PAUSEONERROR=YES

Example

Add the following statement to your CONFIG.SYS file if you do not want OS/2 to pause after the processing of CONFIG.SYS when error messages have been generated.

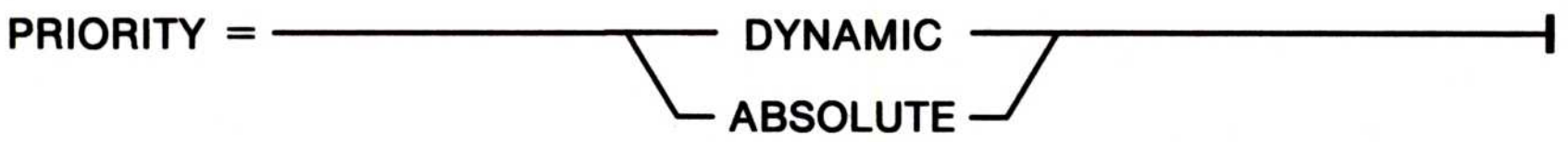
```
PAUSEONERROR=NO
```

PRIORITY

Purpose

Selects priority calculation in scheduling regular class threads in OS/2 mode.

Syntax



Parameters

DYNAMIC Dynamic calculation of priorities.

ABSOLUTE Absolute priorities.

Remarks

OS/2 has a concept of priority classes and priority levels. Regular class threads have their priorities varied dynamically by OS/2. This dynamic change in priority is calculated from a base level, which may be changed by a system call.

Default

PRIORITY = DYNAMIC

Example

You do not want OS/2 to vary the priorities of threads as they are running. Change your PRIORITY statement in CONFIG.SYS:

```
PRIORITY=ABSOLUTE
```

Related Commands

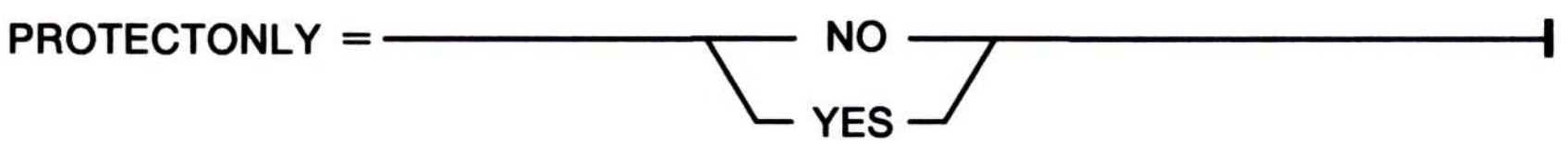
MAXWAIT and TIMESLICE statements in CONFIG.SYS

PROTECTONLY (mode selection)

Purpose

Selects one or two operating modes.

Syntax



Parameters

NO	Both DOS and OS/2 operating modes.
YES	OS/2 mode only.

Remarks

Warning: You should use caution in configuring a system for only OS/2 mode. If you do not have an OS/2 editor and you decide to reconfigure your system to include DOS mode, you will have to rename your CONFIG.SYS file and restart OS/2 so default CONFIG.SYS values can be assigned. (The system editor EDLIN runs only in DOS mode.)

Default

PROTECTONLY = NO

Example

This statement configures your system for OS/2 mode only:

```
PROTECTONLY=YES
```

The following statement configures your system for both OS/2 and DOS operating modes:

```
PROTECTONLY=NO
```

Related Commands

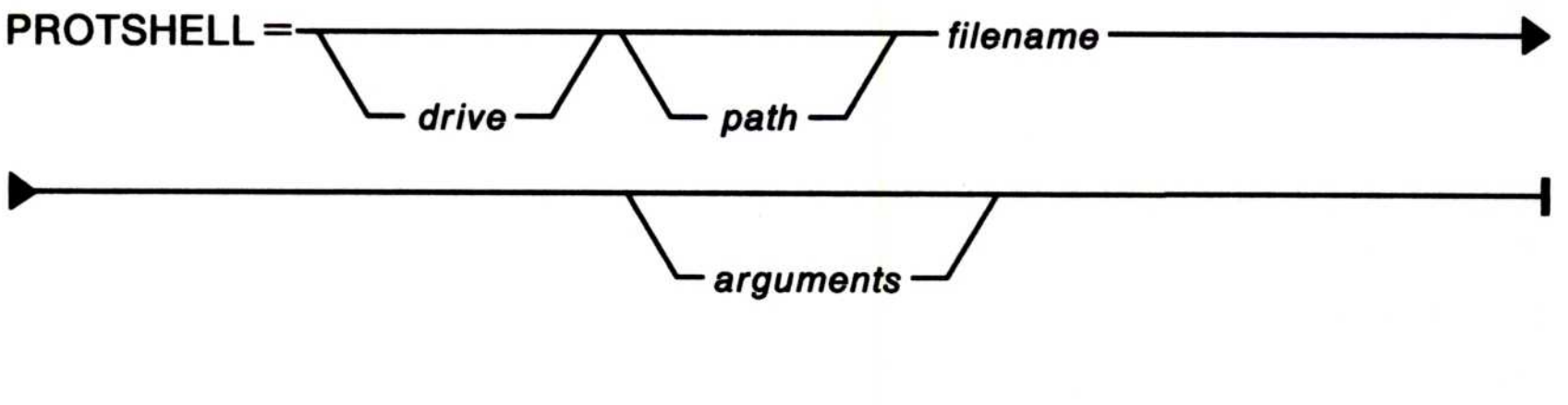
PROTSHELL, RMSIZE, and SHELL statements in CONFIG.SYS

PROTSHELL (user interface)

Purpose

Loads the user interface program and OS/2 command processor.

Syntax



Parameters

filename The complete name of the file, which includes its extension, that contains the user interface.

arguments The parameters required by the user interface. The standard OS/2 user interface (DMPC.EXE) requires the name of the Program Selector configuration file, the name of the Program Selector program file, and the name of the OS/2 mode command processor.

Remarks

If PROTSHELL is not specified in CONFIG.SYS, the default user interface, the OS/2 Program Selector, is loaded and started with an argument of CMD.EXE.

For a system that is configured for both OS/2 and DOS modes, the user interface is started and automatically displayed. The DOS command processor is also started but does not begin running until the DOS command prompt is selected from the Program Selector.

This statement in CONFIG.SYS lets you replace the default OS/2 command processor CMD.EXE with another command processor.

PROTSHELL (user interface)

Default

PROTSHELL = DMPC.EXE SHELL11F.CNF SHELL11F.EXE CMD.EXE

Note: The installation program places this statement in your CONFIG.SYS file:

```
PROTSHELL=DMPC.EXE SHELL11F.CNF SHELL11F.EXE CMD.EXE /K OS2INIT.CMD
```

Example

You have your own command processor for OS/2 mode. Assume it is called MYPROC.EXE and is located in the C:\PROGRAMS directory. Include the following statement in CONFIG.SYS to be able to use your command processor in OS/2 mode:

```
PROTSHELL=DMPC.EXE SHELL11F.CNF SHELL11F.EXE C:\PROGRAMS\MYPROC.EXE
```

Related Commands

PROTECTONLY, RMSIZE, and SHELL statements in CONFIG.SYS

REM (remarks)

Purpose

Allows you to add remarks to your CONFIG.SYS file.

Syntax



Parameters

comment Any text you wish to appear in your CONFIG.SYS file, which will be ignored by the system during the processing of CONFIG.SYS.

Remarks

In addition to using REM statements to add remarks to your CONFIG.SYS file, you can also use them without comments to provide spacing for readability.

Example

This example shows a portion of a CONFIG.SYS file where REM statements have been used to explain system configuration and improve readability.

```
REM  The following statement sets the size of
REM  DOS mode to 512K bytes.
REM
RMSIZE=512
REM
REM  This allows Ctrl+Break in DOS mode.
REM
BREAK=ON
REM
```

RMSIZE (DOS mode size)

Purpose

Specifies the highest storage address allowed for the DOS operating environment.

Syntax



Parameters

x	A number from 0 through 640, representing a multiple of 1024 bytes.

Remarks

If you specify a number in the RMSIZE statement that exceeds the amount of storage available below 640Kb, the system automatically adjusts this number. When you want the largest usable size for DOS mode, you can allow the system to calculate this by accepting the system default; that is, by not putting a RMSIZE statement in CONFIG.SYS.

Default

The default is the lesser of these two:

- The total storage installed minus 512Kb
- 512Kb or 640Kb (whichever is installed in the lower address base).

Example

Suppose you want the size of DOS mode to be 640Kb. Assuming you have enough storage, you can include this statement in CONFIG.SYS:

```
RMSIZE=640
```

Related Commands

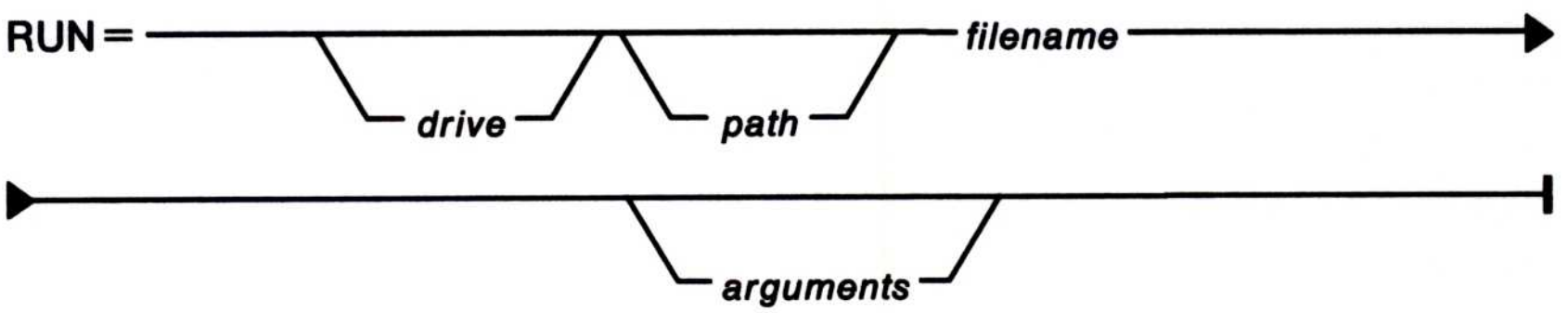
PROTSHELL, PROTECTONLY, and SHELL statements in CONFIG.SYS

RUN

Purpose

Loads and starts a system program during system initialization.

Syntax



Parameters

filename The complete name of the file, which includes its extension, that contains the system program you want to run.

arguments Information passed to the program.

Remarks

A system program that is loaded with a RUN statement is started during system startup. It may only issue I/O requests to the keyboard, mouse, or display after it has established a popup screen.

Note that if a program started with a RUN statement requests a popup screen before the user interface is started, it receives a unique error code. The program may respond by delaying its processing until the user interface is started.

A program started with a RUN statement may establish a keyboard or mouse monitor for any nondetached program.

Since RUN programs are started before initialization of the user interface and disk error handling, the program must disable OS/2 disk error handling and perform its own disk error handling.

More than one RUN statement may appear in CONFIG.SYS. RUN statements are processed in the order in which they appear in the file. All DEVICE statements are processed before any RUN statements.

RUN

Example

You can use the RUN statement to start the print spooler at system startup. Suppose you add this statement to your CONFIG.SYS file:

```
RUN=C:\OS2\SPOOL.EXE
```

After system initialization is complete, you receive a message informing you that the print spooler is running and that data will be printed on the default printer, LPT1. Before you use the RUN statement to start the print spooler, be sure to read the information on the SPOOL command on page 2-126. If you have more than one printer, you can use more than one RUN statement to start more than one print spooler.

Related Commands

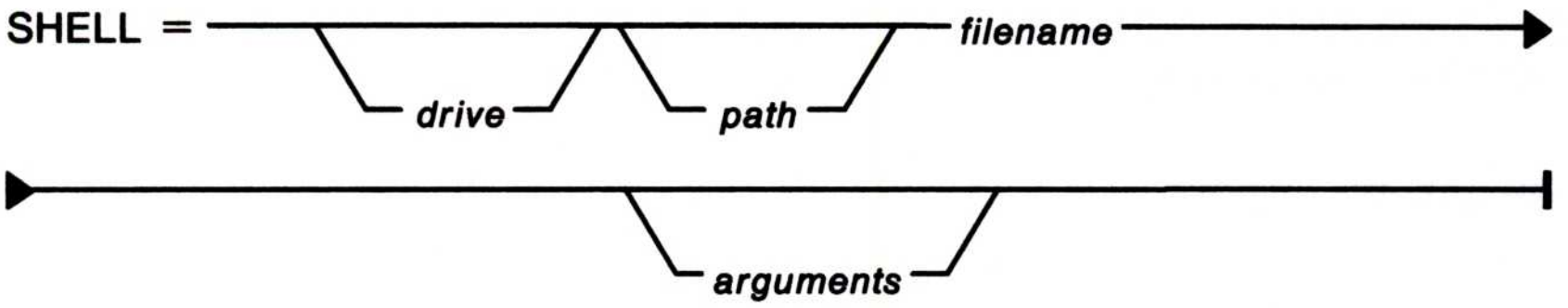
DETACH

SHELL (DOS command processor)

Purpose

Loads and starts the command processor for DOS mode.

Syntax



Parameters

filename The complete name of the file, which includes its extension, that contains the DOS mode command processor.

arguments Information passed to the command processor.

Remarks

This command lets you replace the DOS command processor with another command processor. Substitute the complete name of the file containing your command processor for the *filename* in the syntax. Because the internal commands and the batch processor reside in COMMAND.COM, these functions will not be available to you unless they are duplicated in your command processor.

If SHELL is omitted from the CONFIG.SYS file, the default DOS command processor is loaded and started with an argument of /P. The /P parameter makes the DOS processor permanent in storage, so that you can always select it from the Program Selector.

SHELL does not affect the OS/2 SET command, nor the SHELL command in BASIC.

If you specify a SHELL statement that contains a file specification other than the default, you should set the COMSPEC environment variable to the same file specification when DOS mode is started. This allows the system to find the transient portion of the DOS mode command

processor in the same file as the resident portion. You may do this by placing a SET command in your AUTOEXEC.BAT file. The SET command should specify the same file specification that is in your SHELL statement as the value for COMSPEC. As an alternative, you can enter this SET command at the DOS mode command prompt. See the example below.

Default

SHELL = COMMAND.COM /P

Example

You want to use your own command processor for DOS mode instead of the COMMAND.COM command processor that comes with OS/2. Your command processor is called DOSPROC.COM, and it is located in the C:\DOSPGMS directory. Include the following statement in your CONFIG.SYS file:

```
SHELL=C:\DOSPGMS\DOSPROC.COM
```

Then add the following command to your AUTOEXEC.BAT file:

```
SET COMSPEC=C:\DOSPGMS\DOSPROC.COM
```

The next time you start OS/2, it will load your command processor.

Related Commands

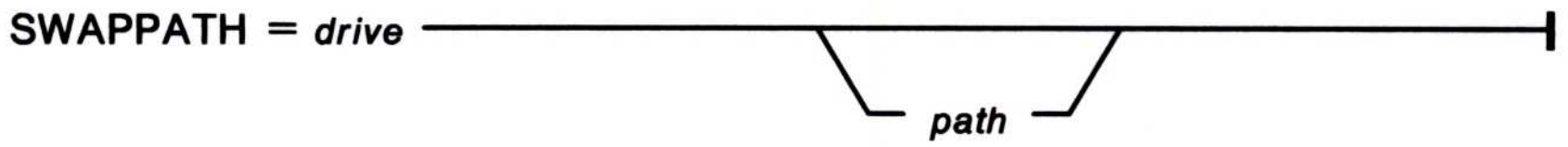
PROTSHELL and RMSIZE statements in CONFIG.SYS

SWAPPATH

Purpose

Specifies the location of the file that keeps track of storage segments that are swapped out in OS/2 mode.

Syntax



Remarks

The swap file needs to be at least 512Kb in size, so you must have at least 512Kb of available storage on the swapping drive. The size of the swap file increases as OS/2 overcommits storage. If you are concerned about the swap file using all available disk space, you can create a separate partition for the swap file. Having the swap file reside in a separate partition can also improve swapping performance.

Note: If your swap file resides on diskette, the diskette must not be removed from the diskette drive.

Default

SWAPPATH=C:\ if starting from a fixed disk
SWAPPATH=A:\ if starting from a diskette

Example

In this example, you have specified MEMMAN=SWAP,MOVE in your CONFIG.SYS file. You would like the swap file stored in the C:\SWAP directory. Use this statement in CONFIG.SYS:

```
SWAPPATH=C:\SWAP
```

Related Commands

MEMMAN statement in CONFIG.SYS

THREADS

Purpose

Determines the maximum number of independent actions, known as threads, for OS/2 mode.

Syntax



Parameters

x	A number from 32 through 255.

Remarks

This command specifies the maximum number of independent actions, known as threads, that may exist in the system at one time. The default value includes an allowance for 24 system threads to be used for OS/2 itself and 40 for application programs. Any system extensions, additional applications, or complex operating environments that require threads must be accounted for here.

Default

THREADS=64

Example

Your system needs to be able to handle up to one hundred active threads. Add the following to CONFIG.SYS:

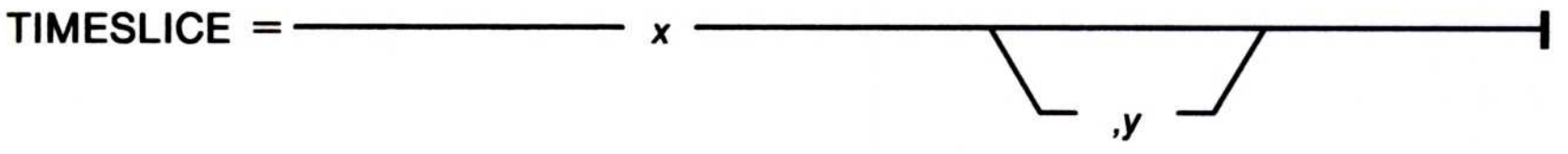
```
THREADS=100
```

TIMESLICE

Purpose

Sets the amount of processor time allocated to processes and programs for both modes.

Syntax



Parameters

x Selects the minimum TIMESLICE value in milliseconds. This value must be an integer greater than or equal to 32.

y Selects the maximum TIMESLICE value in milliseconds. This value must be an integer greater than or equal to the minimum value, and less than 65536.

Remarks

If only one value is specified with TIMESLICE, both minimum and maximum TIMESLICE values are set to that value.

The minimum value is used by OS/2 for special situations and should generally be allowed to default.

The maximum value is the amount of time used for a round robin method of scheduling among threads of equal priority. The default value of 248 milliseconds, or about one quarter of a second, is appropriate for programs where timing is not critical. For programs performing tasks that require precise timing (for example, communications or process control), a lower value such as 100 may be needed to avoid undesirable results. Very small values will lead to excessive processing by OS/2 to support user function. Very large values will effectively disable the round robin scheduling, allowing unfair distribution of system resources.

TIMESLICE

Default

TIMESLICE = 32,248

Example

You are using a communications program that is very timing dependent. The instructions that came with the program tell you to change the maximum time slice allowed to 120 milliseconds. Add the following statement to CONFIG.SYS:

```
TIMESLICE=32,120
```

Related Commands

PRIORITY and MAXWAIT statements in CONFIG.SYS

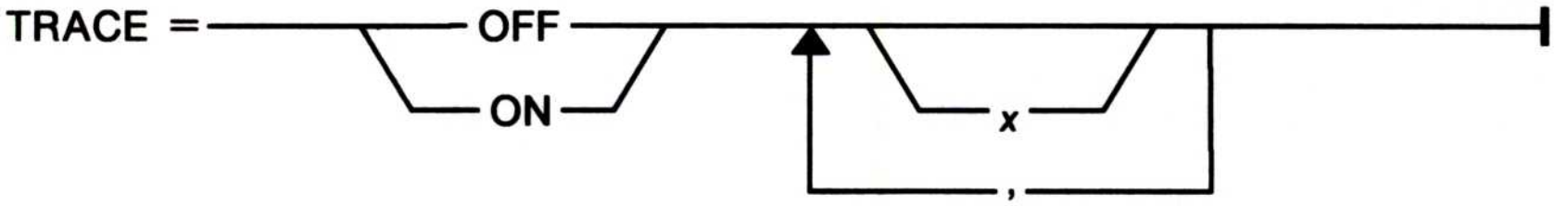
TRACE

Purpose

Selects system trace.

Note: This statement is intended for use with aid from an IBM Service Representative. For more information, see "Commands for Problem Determination" on page 2-6.

Syntax



Parameters

ON	Permits system tracing of specified events.
OFF	Prevents system tracing of specified events.
x	A number from 0 through 255, indicating the major event code to be traced or *not* to be traced. The numbers for these codes will be supplied by your IBM Service Representative.

Remarks

Events are traced in both OS/2 and DOS modes. If no events are listed after the ON or OFF parameter, then ON indicates all events will be traced and OFF indicates no events will be traced.

TRACE statements are processed in the order in which they appear in CONFIG.SYS. Their effects are cumulative. If any part of a statement is incorrect, the statement is ignored.

TRACE

Default

If TRACE is not specified in CONFIG.SYS, no events will be traced as a result of the processing of CONFIG.SYS. However, if a TRACEBUF statement has been specified in CONFIG.SYS, this allocates a trace buffer, and you can trace events by entering the TRACE command at the OS/2 mode command prompt.

If neither TRACE nor TRACEBUF is specified in CONFIG.SYS, then no trace buffer is allocated and system tracing is not available.

Related Commands

TRACEBUF statement in CONFIG.SYS, TRACE, TRACEFMT

TRACEBUF (trace buffer)

Purpose

Sets the size of the circular trace buffer.

Note: This statement is intended for use with aid from an IBM Service Representative. For more information, see "Commands for Problem Determination" on page 2-6.

Syntax



Parameters

x	A number from 1 through 63, representing a multiple of 1024 bytes.

Default

If any valid TRACE statements are in CONFIG.SYS (including TRACE=OFF), the default size of the trace buffer is 4Kb.

If neither TRACE nor TRACEBUF is specified in CONFIG.SYS, then no trace buffer is allocated and system tracing is not available.

Example

In this example, your IBM Service Representative has advised you to increase the size of the trace buffer to 10Kb. Include this statement in your CONFIG.SYS file:

```
TRACEBUF=10
```

Related Commands

TRACE statement in CONFIG.SYS, TRACE, TRACEFMT

Chapter 5. Making the Command Processor Work for You

Certain symbols, referred to as command operators, can be interpreted by the Operating System/2™ (OS/2™) command processors. The symbols and their functions are:

Symbol	Function
<	Redirect input to a command or program
>	Redirect output of a command or program
>>	Append redirected output of a command to a file
¦	Pipe output of one command to another
&& and ¦¦	Conditionally process commands
& and ()	Separate and group commands

Some of these command operators work only in OS/2 mode. In these instances, the OS/2 mode symbol appears beneath the heading of the section that describes them.

Redirecting Input and Output

Redirection symbols allow a program or command to receive its input from a source other than the standard input device (the keyboard or mouse), or direct its output to a device other than the standard output device (the display).

Note: In DOS mode, any output sent to the standard error device cannot be redirected. However in OS/2 mode, it can. See "Redirection in OS/2 Mode" on page 5-5.

The symbol to redirect standard output is the > symbol. For example, if you enter:

```
DIR >PRN
```

the output of DIR, which would normally have gone to the display, is redirected to the printer.

You can also enter:

```
DIR >DIRLIST
```

and redirect DIR's output to a file. If the file does not exist, it is created for you. If the file does exist, it is opened and its contents are overwritten.

The symbols >> also redirect standard output, but append it to existing data. For example, suppose you enter:

```
DIR >>FILE1
```

If FILE1 does not exist, it is created. If it does exist, FILE1 is opened and the write pointer is positioned at the end of the file so that the output of DIR is appended to the contents of the file.

The symbol to redirect standard input is the < symbol. In the following example, the program MYPROG receives its input from the file called INPUT.TXT, instead of from the keyboard:

```
MYPROG <INPUT.TXT
```

Warning: When using this method of providing input to a program, be sure *all* the program's input is in the file. If the program attempts to obtain more input after it reaches end-of-file, OS/2 is unable to supply the input and stops processing.

Piping Output to Another Command

The OS/2 piping symbol (ASCII character 124) allows the standard output of one program to be used as the standard input to another program. In OS/2 mode, OS/2 uses a storage buffer mechanism to hold the input and output data being piped. In DOS mode, OS/2 uses a temporary file to hold the input and output data being piped.

Note: The piping symbol (ASCII character 124) is shown on the keyboard key top as a broken vertical bar (¦). If your system has not been set up for code page switching, this is the character that is displayed and printed. However, if your system has been set up for code page switching, the piping symbol will appear as an unbroken vertical bar (|).

The piping symbol can be used with programs or commands that receive their input from the standard input device, modify the data, and then send the changed data to the standard output device. These

commands, or programs, are referred to as *filters*. OS/2 has three such filter commands, described in Chapter 2. They are:

SORT Sorts text data.

FIND Searches files for occurrences of specified strings of text.

MORE Displays data, a full screen at a time, then pauses with the message "--More--."

Here are some examples using the piping symbol with OS/2 filters and redirection symbols. First, an example using just the piping symbol: suppose you enter:

```
DIR ¦ SORT
```

This generates a sorted directory listing. All standard output generated by the DIR command is sent to the standard input of SORT.

To put the sorted directory in a file, enter:

```
DIR ¦ SORT > FILE
```

If you want the file to contain only the directory entries for subdirectories, enter:

```
DIR ¦ FIND "<DIR>" ¦ SORT > FILE
```

Processing Commands Conditionally



Command operators allow you to make the processing of commands dependent on the success or failure of other commands.

The AND operator (&&) allows a command to run only if the one that precedes it is successful. That is, if the command on the left side of the && is successful, the command on the right side is performed. If the command on the left side of the && fails, then the command on the right side is not performed.

For example, if you were to enter:

```
DIR LETTER && TYPE LETTER
```

the file LETTER is displayed on the screen by TYPE only if it exists in your current directory.

When you use the OR operator (ASCII character 124, entered twice) with commands, this allows a command to run only if the one that precedes it fails. That is, if the command on the left side of the OR is not successful, the command on the right side is performed. If the command on the left side of the OR is successful, the command on the right side is not performed.

Note: If code page switching is in effect on your system, the OR command operator appears as two unbroken vertical bars (||) when entered. If code page switching is not in effect, the OR command operator appears as two broken vertical bars (¦¦). You can enter the OR command operator by pressing the piping symbol key (¦) twice.

For example, suppose you enter:

```
DEL MEMOS ¦¦ RMDIR MEMOS
```

the file MEMOS is deleted, if it exists in your current directory. Only if a file MEMOS does not exist, is the MEMOS directory removed, if it exists.

Separating and Grouping Commands

The command separator symbol (&) performs the commands on both sides of the symbol from left to right and produces the output from both commands after the second command is run. For example, if you were to enter:

```
DIR C:*.BAT & DIR D:*.CMD
```

this displays the directory of files with an extension of .BAT in drive C followed by the directory of files with an extension of .CMD in drive D.

Internal commands can be grouped by using parentheses. Parentheses ensure running of commands in the proper order. For example,

```
DIR PLAN && TYPE PLAN ¦ SORT > SORTPLAN
```

produces the same results as:

```
DIR PLAN && (TYPE PLAN ¦ SORT > SORTPLAN)
```

Both statements check to see if the file PLAN exists, and if it does, they sort the file and put the results in a file called SORTPLAN. However, the following example,

```
(DIR PLAN && TYPE PLAN) ¦ SORT > SORTPLAN
```

sorts the directory entry for PLAN as well as the contents of the PLAN file.

If you are using parentheses and you leave off the closing parenthesis, OS/2 displays the message:

```
More?
```

You can respond by either finishing the command with the closing parenthesis and pressing Enter, or you can press Ctrl+Break to cancel the command.

Redirection in OS/2 Mode



In OS/2 mode, before a command is issued, its input and output may be redirected to files or other devices using digits (the numbers 0 through 9) in redirection sequences that are then interpreted by CMD.EXE. These digits are used as internal file IDs by CMD.EXE and also by OS/2 application programs. When used in redirection sequences with a command, input or output from or to an internal file ID is received or obtained from the newly substituted device or file. At the completion of the command containing the redirection sequence, input or output of an internal file ID is no longer redirected.

CMD.EXE interprets the digits 0 through 2 as:

0	standard input
1	standard output
2	standard error

A program's input and output can be redirected only if the program is reading from standard input and writing to standard output, including standard error. If the program or command is reading directly from the keyboard and writing directly to the display by means of function calls, then its output cannot be redirected.

Digits 3 through 9 are not normally used by CMD.EXE or application programs. However, input or output of any internal file ID consisting

of a single-digit number can be redirected. Redirection takes place whether or not the internal file ID is currently being used by another application program.

Redirection Sequences

The redirection sequences shown below may appear anywhere in a simple command or within the body of one of the special commands IF, FOR, or DETACH. The *n* represents digits 0 through 9, and *name* specifies the name of a file or device.

Redirection is performed in the order in which it appears on the command line; therefore this order is important. Substitution of environment and batch file variables both occur prior to using the name of the device or file appearing in the sequence to perform the redirection.

Sequence	Interpretation by CMD.EXE
<name	Use file *name* as standard input (internal file ID 0). If the file doesn't exist, an error occurs.
>name	Use file or device *name* as standard output (internal file ID 1). If the file doesn't exist, it is created; otherwise it is truncated to zero length.
>>name	Use file *name* as standard output (internal file ID 1). If the file exists, output is appended to the end of the file; otherwise the file is created.
<&n	Standard input (internal file ID 0) is duplicated from the internal file ID *n*.
>&n	Standard output (internal file ID 1) is duplicated from the internal file ID *n*. **Note:** If any of the above sequences is preceded by *n*, then the internal file ID redirected is the one specified by the first *n* instead of the default 0 or 1, as follows:
2>&1	Standard error (internal file ID 2) becomes a duplicate of standard output (internal file ID 1). Any output written by a process to internal file ID 2 has the same effect as if it had been sent to internal file ID 1.

4>*name* Internal file ID 4 becomes the file specified by *name*. Any output written by a process to internal file ID 4 is sent instead to this file.

Note that no other characters, including blanks, may exist between the first *n* (if used), the redirection indicator, the append indicator (if used), and the & character (if used). Blanks may appear only between the redirection or append indicators and their related *names* or the & character and its related *n*.

To be recognized, a leading or first *n* must be seen to be the start of a new redirection sequence. To qualify, it must be the first character on a line, or it must be immediately preceded by a blank or one of the following delimiters:

```
& | ; , ( ) = "
```

Otherwise it is interpreted by CMD.EXE as part of the previous redirection sequence. The redirection symbol following it is then interpreted as the beginning of the redirection sequence. Redirection commands cannot be enclosed in quotes.

Examples of Redirected I/O

The following examples assume that the programs use standard input, standard output, and standard error.

Example 1

```
COBOL PROGRAM.COB >OUTLOG;2>ERRLOG
```

If the COBOL compiler sends its output to standard output and its errors to standard error, then the output is sent to the file OUTLOG and the error report, to the file ERRLOG.

Example 2

```
(CHKDSK *.* >FILELOG)2>&1
```

This statement directs all output (standard output and standard error) to be sent to the file FILELOG. Its equivalent is:

```
CHKDSK *.* >FILELOG 2>&1
```

Example 3

```
DETACH >DATALOG APP 2>&1
```

This statement is equivalent to:

```
DETACH APP >DATALOG 2>&1
```

Both statements indicate that the program APP is executed as a noninteractive process whose standard output is sent to the file DATALOG, and any output written to internal file ID 2 (standard error) is redirected to the current target of internal file ID 1; in this case, DATALOG.

Example 4

```
ECHO "HELLO"2>>OUTFILE
```

This statement echoes the character string "HELLO" to the screen and appends standard error output to the file OUTFILE. Note that if the character string were not enclosed in quotes but entered like this:

```
ECHO HELLO2>>OUTFILE
```

the output of the character string "HELLO2" would be appended to the file OUTFILE.

Example 5

```
DIR XYZ || DIR XYZ 2>ERRLOG
```

If the file XYZ does not exist, the error output of the first DIR command goes to the screen, and the output of the second DIR is sent to the file ERRLOG.

Example 6

```
>OUTFILE (IF EXIST MAKEFILE ECHO "It exists")
```

The character string "It exists" is written to the file OUTFILE if the file MAKEFILE exists.

Note: Only one redirection command for each internal file ID is performed. Therefore, if you were to type:

```
DIR >FNAME1 >FNAME2 >FNAME3 >FNAME4
```

the four redirection operations are screened following parsing and duplicates are removed, with the last one taking precedence. The output is placed in FNAME4, with no creation of the other files specified.

Running Noninteractive Programs

Redirecting standard error as well as standard output allows no screen output to occur. By using redirection sequences with the DETACH command, you can cause programs that use both standard output and standard error to process completely noninteractively.

```
DETACH APP >OUTLOG 2>ERRLOG
```

In this example, no standard screen output occurs.

Since redirection is processed from left to right on the command line, order is *important*. To redirect both standard output and standard error to a single file, the command must use the following order:

```
command arguments >FNAME 2>&1
```

Using the above syntax to direct all output to a single file, enter, for example:

```
DETACH APP >OUTLOG 2>&1
```

Preventing All Output by a Program

It is possible to redirect any internal file ID from 0 through 9 for input or output. Files opened as input internal file IDs are opened read-only, whereas files opened as output internal file IDs are either write-only (for normal redirection) or read-write (for appending). No attempt is made by CMD.EXE to detect standard output internal file IDs redirected as input or vice-versa. Thus by redirecting both standard output and standard error as input internal file IDs, as in for example:

```
command arguments 1<INFILE 2<INFILE
```

it is possible to end up with no output at all. All output can also be prevented by redirecting standard output and standard error to NUL as follows:

```
command arguments >NUL 2>NUL
```

Echoing of Redirection Statements in OS/2 Mode

Since the two default internal file IDs can be redirected (internal file ID 0 for standard input and internal file ID 1 for standard output), it is necessary for you to visually determine which internal file IDs are being manipulated. For this reason, all statements containing redirection sequences are echoed by CMD.EXE in a format where the internal file ID number precedes the redirection symbol. For example, if you run a batch file (with the extension .BAT or .CMD) that contains the following command:

```
DIR >DIRFILE
```

the following is echoed to the screen:

```
DIR 1>DIRFILE
```

If your batch file contains:

```
MORE <DIRFILE
```

when it runs, CMD.EXE echoes:

```
MORE 0<DIRFILE
```

Allowing Symbols to be Input as Text



The escape symbol (^) allows input of normally significant characters as text. When used outside of quoted strings during command line input, the ^ symbol causes the next character following it to be treated as normal text. Here is an example of its effect when you use it with the ECHO batch command. If you type:

```
ECHO HELLO ^>FILE
```

this is what is displayed by ECHO:

```
HELLO >FILE
```

Appendix A. A Quick Reference to OS/2 Commands

This appendix provides an overview of Operating System/2™ (OS/2™) commands. The syntax for a particular command can be found on the page listed in the "purpose" column.

Commands for One Mode Only

OS/2 has two modes of operation:

- OS/2
- DOS.

There are some commands that provide functions specific to one mode. In this book, these symbols identify commands that work in only one mode:

OS/2 only



DOS only



If you enter a command that has a symbol for one mode while you are in the other mode, an error message is displayed.

Note: There are some IBM Personal Computer DOS Version 3.3 commands that are not supported or have changed slightly for DOS mode. See Appendix B, "DOS Compatibility" on page B-1.

OS/2 Commands

Command	Purpose
ANSI 	Enables or disables extended display and keyboard support in OS/2 mode (page 2-7).
APPEND 	Locates data files outside of the current directory (page 2-8).
ASSIGN 	Assigns a drive letter to a different drive (page 2-12).
ATTRIB	Turns on or off the read-only and archive attributes of a file (page 2-14).
BACKUP	Backs up one or more files from one disk to another (page 2-17).
BREAK 	Sets on or off the check for Ctrl+Break in DOS mode whenever a program issues system calls (page 2-20).
CHCP	Displays or changes the current system character set (page 2-22).
CHDIR	Changes the current directory or displays its name (page 2-26).
CHKDSK	Scans the disk and checks it for errors (page 2-28).
CLS	Clears the display screen (page 2-31).
CMD 	Starts another OS/2 command processor (page 2-32).
COMMAND 	Starts another DOS command processor (page 2-34).
COMP	Compares the contents of two files (page 2-36).

Command	Purpose
COPY	Copies one or more files and also combines files (page 2-39).
CREATEDD	Creates a dump data diskette (page 2-43).
DATE	Displays the system date (page 2-45).
DETACH	Starts a noninteractive program in OS/2 mode (page 2-47).
DIR	Lists the files in a directory (page 2-49).
DISKCOMP	Compares the contents of two diskettes (page 2-53).
DISKCOPY	Copies the contents of one diskette to another diskette (page 2-56).
DPATH	Locates data files outside of the current directory (page 2-59).
ERASE	Deletes one or more files (page 2-61).
EXIT	Ends the current command processor (page 2-63).
FDISK	Creates and deletes partitions and logical drives, changes the primary partition, displays partition data, and selects next fixed disk drive for partitioning (page 2-64).
FIND	Searches for a specific string of text in a file (page 2-67).
FORMAT	Formats the disk to accept OS/2 files (page 2-70).
GRAFTABL	Allows the extended character set to be displayed when using display adapters in graphics mode (page 2-74).
HELP	Provides help information related to a warning or error message (page 2-76).

Command	Purpose
JOIN 	Joins a disk drive to a specific path (page 2-79).
KEYB 	Specifies a special keyboard layout that replaces the default U.S. keyboard layout (page 2-81).
LABEL	Creates, changes, or deletes the volume identification label on a disk (page 2-84).
MKDIR	Creates a new directory (page 2-86).
MODE	Sets operational modes for devices (page 2-88).
MORE	Sends output from a file to the screen one full screen at a time (page 2-97).
PATCH	Allows you to apply IBM supplied patches (page 2-98).
PATH	Locates commands outside of the current directory (page 2-101).
PRINT	Prints or cancels printing of one or more files (page 2-103).
PROMPT	Set the OS/2 command prompt (page 2-105).
RECOVER	Recovers files from a disk containing defective sectors (page 2-107).
RENAME	Changes the name of a file (page 2-109).
REPLACE	Selectively replaces files (page 2-110).
RESTORE	Restores one or more backup files from one disk to another (page 2-112).
RMDIR	Removes a directory (page 2-115).
SET	Sets one string value in the environment equal to another string for later use in programs (page 2-117).

Command	Purpose
SETCOM40 	Sets the COM port for DOS mode (page 2-121).
SORT	Reads data from standard input, sorts it, and writes it to standard output (page 2-124).
SPOOL 	Intercepts and separates data from different sources going to the printer so printer output is not intermixed. (page 2-126).
START 	Starts an OS/2 program in another session. Can be used in STARTUP.CMD autostart facility (page 2-130).
SUBST 	Substitutes a drive letter for another drive or path (page 2-133).
SYS	Transfers the OS/2 hidden files IBMBIO.COM and IBMDOS.COM from one diskette or fixed disk to another diskette or fixed disk. (page 2-135).
TIME	Displays and sets the system time (page 2-137).
TRACE 	Selects system trace (page 2-139).
TRACEFMT 	Displays formatted trace records in reverse time stamp order (page 2-141).
TREE	Displays all of the directory paths (page 2-143).
TYPE	Displays the contents of a file (page 2-144).
VER	Displays the OS/2 version number (page 2-146).
VERIFY	Confirms that data written to a file is correct (page 2-147).
VOL	Displays the disk volume label (page 2-149).
XCOPY	Selectively copies groups of files, including those in subdirectories, from one disk to another (page 2-151).

Batch Commands

Command	Purpose
CALL	Nests a batch file within a batch file (page 3-11).
ECHO	Allows or prevents display of OS/2 commands while a batch file is running (page 3-13).
ENDLOCAL 	Restores the drive, directory, and variables that were in effect before a SETLOCAL command was issued (page 3-15).
EXTPROC 	Defines an external batch file processor (page 3-17).
FOR	Allows repetitive processing of commands within a batch file (page 3-18).
GOTO	Transfers control elsewhere in the batch file (page 3-20).
IF	Allows conditional processing of commands within a batch file (page 3-21).
PAUSE	Suspends processing of the batch file (page 3-24).
REM	Displays remarks from within a batch file (page 3-26).
SETLOCAL 	Sets the drive, directory and variables that are local to the current batch file (page 3-27).
SHIFT	Allows more than 10 replaceable parameters in a batch file (page 3-28).

Statements for the CONFIG.SYS File

Statement	Purpose
BREAK 	Turns on or off the check for Ctrl+Break during program processing in DOS mode (page 4-5).
BUFFERS	Sets the number of disk buffers for use (page 4-7).
CODEPAGE	Selects system character sets to be prepared by OS/2 for character set switching (page 4-8).
COUNTRY	Sets the country dependent information (page 4-10).
DEVICE	Specifies the path and filename of a device driver to be installed (page 4-13).
DEVINFO	Prepares a device for character set switching (page 4-30).
DISKCACHE	Assigns storage to be used for disk cache (page 4-34).
FCBS 	Sets File Control Block information for DOS mode (page 4-36).
IOPL 	Enables I/O privilege to be granted or denied to requesting processes in OS/2 mode (page 4-38).
LIBPATH 	Identifies the locations of dynamic link libraries in OS/2 mode (page 4-39).
MAXWAIT 	Sets time limit for lack of access to processor resource in OS/2 mode (page 4-41).
MEMMAN 	Selects storage allocation options for OS/2 mode (page 4-42).

Statement	Purpose
PAUSEONERROR	Enables or disables pausing when error or warning are issued during the processing of CONFIG.SYS (page 4-44).
PRIORITY 	Selects priority calculation in scheduling regular class threads in OS/2 mode (page 4-45).
PROTECTONLY	Specifies an OS/2-only operating evironment (page 4-46).
PROTSHELL 	Specifies the user interface program and the OS/2 command processor (page 4-47).
REM	Allows you to add remarks to your CONFIG.SYS file (page 4-49).
RMSIZE 	Sets the amount of storage used for DOS programs (page 4-50).
RUN	Loads and starts a system program during system startup (page 4-51).
SHELL 	Loads the command processor for DOS mode (page 4-53).
SWAPPATH 	Identifies the location of the file that keeps track of segments swapped out of storage in OS/2 mode (page 4-55).
THREADS 	Determines the maximum number of independent actions, known as threads, in OS/2 mode (page 4-56).
TIMESLICE 	Sets amount of processor time each process receives (page 4-57).
TRACE	Selects system trace (page 4-59).

Statement	Purpose
TRACEBUF	Sets the size of the system trace buffer (page 4-61).

Appendix B. DOS Compatibility

This appendix discusses compatibility between DOS mode of Operating System/2™ (OS/2™) and IBM Personal Computer DOS Version 3.3.

Compatibility Exceptions for Commands

There are some IBM Disk Operating System commands that are either not supported or whose syntax has changed slightly for DOS mode. The table below provides you with an overview of these commands and their differences.

IBM DOS	DOS Mode in OS/2
APPEND	/X parameter is rejected.
COMP	End-of-file checking has been removed. User may continue comparing if file sizes differ.
COUNTRY	Statement specifying code page file path is ignored in CONFIG.SYS.
CTTY	This command is not supported.
DISKCOMP	/1 and /8 parameters are not required because DISKCOMP determines the number of sides and sectors by the source diskette.
DISKCOPY	The /1 parameter is not required because DISKCOPY determines the number of sides by the source diskette.
EXE2BIN	This command is not supported.
FILES	This statement is ignored in CONFIG.SYS.
FORMAT	/1, /8, and /B parameters are not supported. The /S parameter creates more system files than the standard three for IBM DOS.

IBM DOS	DOS Mode in OS/2
GRAPHICS	This command is not supported.
KEYB	Supported in OS/2 mode, not DOS. Code page path parameter is ignored.
LABEL	Option to delete label is not supported.
LASTDRIVE	This statement is ignored in CONFIG.SYS.
MODE	• DISPLAY: Shifting of screen, test pattern not supported • ASYNC: Infinite retry (,P) is emulated through an extended time out. • Code page options are not supported, since this function is now enhanced by CODEPAGE and DEVINFO statements in CONFIG.SYS and the CHCP command. • Redirection of parallel to serial support is not supported in MODE, since this function is now in the SPOOL command. • Ctrl+ Break function deleted.
NLSFUNC	This command is not supported because code page support is internal to OS/2.
PRINT	/B, /U, /M, /S, and /Q parameters are not supported.
SELECT	This command is not supported, since its function is internal to the System Installation program.
SHARE	This command is not supported, since its function is internal to the OS/2 file system.
SYS	Transferring system files to a program diskette is not supported. Target diskette must be a newly formatted diskette, or one that contains IBMBIO.COM and IBMDOS.COM system files.

Guidelines for Running DOS Programs

Many existing application programs written for IBM Personal Computer DOS will run in DOS mode of Operating System/2. Those programs that may *not* run in DOS mode include timing-dependent programs (such as communications and real time programs), hardware-specific routines (such as device drivers), and network-dependent programs.

To display the amount of storage available to run your application programs, use the CHKDSK command while in DOS mode. For more information on CHKDSK refer to page 2-28.

While running in DOS mode you can access any of your data files created by your IBM Personal Computer DOS programs. It is not necessary to reformat your fixed disk or back up and restore files, because system installation replaces only the necessary DOS files on the root directory of your fixed disk, leaving your other files intact.

Running Batch Files

Existing batch files can run unchanged in DOS mode. Batch processing in OS/2 mode requires a file extension of .CMD.

System installation creates an AUTOEXEC file for you, which contains a PATH command and a CALL command. Before you add the commands you need for your system, refer to the discussion, "AUTOEXEC.BAT File" on page 3-5.

Installing and Running Copy-Protected DOS Application Programs

Some copy-protected applications are dependent on the operating system or a version of the operating system for correct installation. Refer to your application manual for specific instructions on which operating system is supported before installing or deinstalling on OS/2.

Always install and deinstall these applications from the DOS command prompt, not from the OS/2 command prompt.

Copy-protected applications may already be installed on your fixed disk. Once installed, many of these applications can run in DOS mode.

Some copy-protected applications rely on timing during their installation and initial startup. During installation and initial startup up of these applications, make sure that no other OS/2 application is running and you do not switch out of DOS mode. This will reduce the possibility of an error occurring.

Using the SPOOL Command

Operating System/2 has its own spooler to control information going to the printer from different sources. This allows OS/2 application programs to send output to the print spooler while your DOS mode application program is printing.

Many DOS application programs do not contain instructions required by the print spooler, and output is not printed until the application program ends. To force printer output for a DOS mode application, press Ctrl+Alt+PrtSc.

When your DOS application program prints using the print spooler, more time may be required than when running without the print spooler. If the print spooler has several jobs to print, jobs are printed sequentially. When one job finishes the next one begins.

For more information on the SPOOL command, see page 2-126.

Using Serial Printers and Plotters

Serial printers and plotters receive output from the asynchronous communications port in your personal computer. In DOS the asynchronous support was part of the base system. In OS/2, you can select the asynchronous support your OS/2 applications may need when you configure your system.

When you select serial device support during system installation, a DEVICE=COM0X.SYS statement is included in your CONFIG.SYS file. If you are using an IBM Personal Computer AT, the statement is:

```
DEVICE=COM01.SYS
```

If you are using an IBM Personal System/2, the statement is:

```
DEVICE=COM02.SYS
```

Your existing DOS mode applications are not written to use this asynchronous support; therefore, you need only select serial device support during system installation if you will be running existing DOS mode applications in conjunction with the following:

- Running the OS/2 application programs or system programs such as SPOOL that use asynchronous communications (serial) support.
- Setting asynchronous communications modes through the MODE command.
- Redirecting of parallel printer data (LPT1) to a serial device (COM1).

Setting Asynchronous Communications Modes

The setting of asynchronous communication modes for a serial printer or plotter in DOS mode of OS/2 is the same as for IBM Personal Computer DOS. For example, if you used the following command in DOS, you would enter the same command at the OS/2 command prompt.

```
MODE COM1:12,N,8,1,P
```

You could also enter this command at the DOS prompt; however, it is recommended that all asynchronous communications settings be done in OS/2 mode. For more information on the MODE command, see page 2-88. A DEVICE=COM0X.SYS statement must be in the CONFIG.SYS file for OS/2 to recognize COM1 as a device name.

Redirecting Parallel Printer Output to a Serial Device

In Operating System/2, redirection of parallel printer output to a serial device is supported by the SPOOL command, not the MODE command as in IBM Personal Computer DOS.

If during OS/2 system installation you selected SPOOL with the default LPT1 as your printer device, system installation put this statement in your CONFIG.SYS file:

```
RUN=C:\OS2\SPOOL.EXE /D:LPT1 /O:LPT1
```

This starts the print spooler in a noninteractive session at system startup, and output is printed on LPT1.

Suppose your application program supported a serial printer through the IBM Personal Computer DOS command, MODE LPT1=COM1. To redirect printer output to COM1 in OS/2, you need to change the /O: parameter of this RUN statement. You can use an editor to make the change to this:

```
RUN=C:\OS2\SPOOL.EXE /D:LPT1 /O:COM1
```

Then restart your system so the change takes effect.

For more information on the SPOOL command, refer to page 2-126.

Using Serial Printers and Plotters with DEVICE=COM0x.SYS

If you wish to use a serial printer or plotter in DOS mode, and your CONFIG.SYS file contains a DEVICE=COM0*x*.SYS statement, you may need to use the DOS mode command, SETCOM40. Some existing IBM Personal Computer DOS applications need the address of the COM port to which the serial printer or plotter is attached, to send data to these devices. SETCOM40 provides these applications with this port address. If you do not know whether your application needs this information, then use SETCOM40 in case it does. For example, if your serial printer or plotter is connected to the COM1 port, then you can enter:

```
SETCOM40 COM1=ON
```

This sets the COM port address. To remove the address when your application in DOS mode no longer needs it, type:

```
SETCOM40 COM1=OFF
```

This lets OS/2 again manage the COM1 port.

Managing Applications Using Serial Devices in DOS Mode

Operating System/2 cannot manage IBM Personal Computer DOS applications using serial devices in DOS mode, because these applications have not been written for a multitasking environment. Typically, these applications go directly to a serial device and try to use it, instead of first checking to see if it is in use by another application.

To use the same serial printer and plotter in both OS/2 and DOS mode, you have to manage the use of these devices yourself while in DOS

mode. When running an application in either OS/2 or DOS mode, make sure that no other application uses a serial printer or plotter at the same time. Be sure to let your DOS mode application finish printing or plotting before switching to an OS/2 session; otherwise unpredictable results, such as intermixed output, may occur.

To manage the printing or plotting of your application programs:

- Be sure to add a DEVICE=COM0X.SYS statement to your CONFIG.SYS file.
- Issue the MODE command from the OS/2 command prompt to set your asynchronous communications mode for your serial printer or plotter.
- When an OS/2 application program is using the serial printer or plotter, let it finish printing or plotting before using the same device in DOS mode.
- When OS/2 mode is finished printing:
  1. At the DOS command prompt type:

     ```
     SETCOM40 COM1=ON
     ```

  2. Start the DOS mode application. Use the serial printer or plotter in DOS mode. When the application has finished printing or plotting exit the application.
  3. At DOS command prompt, type:

     ```
     SETCOM40 COM1=OFF
     ```

- If the DOS mode application modified the settings for the asynchronous communications mode, reset them for OS/2 application programs.

Using a Mouse

The particular OS/2 mouse device driver you are using must be specified in your CONFIG.SYS file. If you indicated your mouse type during OS/2 system installation, the correct OS/2 mouse device driver is already specified in your CONFIG.SYS file. For more information on mouse limitations, refer to "System Limitations."

Using Applications With Enhanced Graphics Features

For some applications using an Enhanced Graphics Adapter, the system may not be able to restore the application's graphics screen when you switch from DOS mode and then switch back to it again. If this occurs, do not switch from DOS mode until you have completed your task.

System Limitations

When using DOS mode, some limitations apply.

Application Programs

- Applications compiled with the IBM BASIC: Compiler 1.0 are not supported.
- Applications that use direct hardware control to back up or restore files may not operate correctly.
- Applications that use Alt, Ctrl, Ctrl+Esc, Ctrl+Alt, and Alt+Esc for application functions may give unexpected results. OS/2 uses Alt+Esc, Ctrl+Esc, and Ctrl+Alt+PrtSc to control the display screen and programs.
- Applications that use block device drivers, such as RAM disks and tape systems, will not load or run.
- Applications that communicate through modems or local area networks are not supported.
- Existing mouse device drivers specified in the CONFIG.SYS file with a .SYS extension are not supported. Other existing mouse device drivers not loaded in the CONFIG.SYS file may work.
- Applications may not function properly if they determine storage sizes by checking hardware RAM.
- Applications that use DOS extended storage may not function properly.

Communications

Communications features of applications are not supported.

Networks

Networking features of applications are not supported.

Plotters

Some existing application programs that use plotters and rely on timing may not run correctly.

Print Spoolers

Existing print spoolers are not supported in DOS mode. Application programs must use the OS/2 SPOOL command. If your application does noninteractive printing, make sure that all printing is finished before the application is ended.

Appendix C. The Line Editor (EDLIN)

This appendix describes how to use the system editor, EDLIN. You can use EDLIN to create, change, and display source program and text files. You can also use EDLIN to edit binary files.

Note: EDLIN works only if your system is configured for DOS mode.

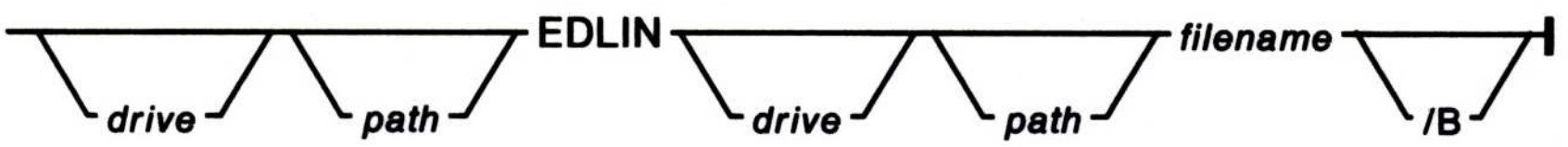
The text of files created or edited by EDLIN is divided into lines of varying length, up to 253 characters per line. Line numbers are generated and displayed by EDLIN during the editing process, but are not actually present in the saved file.

When you insert lines, all line numbers following the inserted text increase automatically by the number of lines inserted. When you delete lines, all line numbers following the deleted text decrease automatically by the number of lines deleted. Consequently, line numbers always go consecutively from 1 through the last line number.

Note: EDLIN erases the original backup copy (.BAK) of the file when you issue an **E** (end edit) command, or if the disk space is required during the editing session to satisfy a **W** (write lines) command.

How to Start the EDLIN Program

Here is the syntax for the EDLIN command:



Editing an Existing File

Enter the name of the file you want to edit, for example:

```
EDLIN MYFILE.TXT
```

If the file you specify exists on the designated or default drive, it is loaded into storage until available storage in DOS mode is 75% full, or until approximately 64K characters have been read, whichever amount is smaller. If a Ctrl+Z character is encountered, which indicates the end of the file, the following message and prompt are displayed:

```
The entire input file has been read.
*_
```

You can now edit the file. Note that the prompt is an asterisk (*).

If the file is too big to fit into available storage, EDLIN loads lines until storage in DOS mode is 75% full, or until approximately 64K characters have been read, whichever amount is smaller. The EDLIN prompt (*) is displayed, and you can then edit the portion of the file now in storage.

To edit the remainder of the file, you must write some of the lines you have edited to disk. You can then load unedited lines from disk into the freed storage. Refer to the **W** (write lines) and **A** (append lines) commands in this appendix for the procedure.

Editing a New File

Specify a file that does not exist on the current drive. A new file is opened with the specified name and the following message and prompt are displayed:

```
EDLIN has created a new file.
*_
```

Begin creating the file by entering the desired lines of text. To begin entering text, enter an **I** (Insert Lines) command to insert lines.

When you have completed the editing session, save the file using the End Edit command. The End Edit command is discussed in this appendix under the heading, "The EDLIN Commands." The new file is saved with the filename and extension you specified in the EDLIN command when you first opened the file.

Note: If you edit an existing file, when you end the edit session with the End Edit command, the original file is saved with an extension of .BAK. You cannot edit a file with a filename extension of .BAK with EDLIN because the system assumes it is a backup file. If it is neces-

sary to edit such a file, rename the file using a different extension; then start EDLIN and specify the new name.

Editing a Binary File

You can use the /B parameter to edit a file that contains embedded Ctrl+Z characters. When you specify /B with the filename of an existing file, EDLIN then loads the entire file regardless of any embedded end-of-file marks.

EDLIN allows you to insert control characters in text, using the **I** (insert lines) command and the Ctrl+V key combination. When you use Ctrl+V, this tells EDLIN that the next character is a control character. You must enter this character in uppercase. For example, to enter the Ctrl+Z character, enter Z after you have pressed Ctrl+V. You can also use this sequence to search for and replace control characters. See the Search and Replace commands.

The EDLIN Command Parameters

Parameter	Definition
line	Denotes when you must specify a line number. There are three possible entries that you can make using this parameter: • Enter a decimal integer from 1−65529. If you specify a number greater than the number of lines in storage, the line is added after the last line. Separate line numbers from each other by a comma or space. • Enter a pound sign (#) to specify the line after the last line in storage. Entering a # has the same effect as specifying a number greater than the number of lines in storage. • Enter a period (.) to specify the current line. The current line indicates the location of the last change to the file, but it is not necessarily the last line displayed. The current line is marked by an asterisk (*) between the line number and the first character of text in the line. For example: `10:*First character of text`

Parameter	Definition
n	Denotes when you must specify lines. Enter the number of lines that you want to write to disk or load from the disk. Use this parameter only with the **W** (write lines) and **A** (append lines) commands. These commands are meaningful only if the file to be edited is too large to fit in storage.
string	Denotes when you must enter one or more characters to represent text to be found, replaced, or deleted, or to replace other text. Use this parameter only with the **S** (search text) and **R** (replace text) commands.

The EDLIN Commands

This section describes the purpose, syntax, and use of the EDLIN commands. The commands are in alphabetic order, each with its purpose, format, and remarks. Examples are provided where appropriate. The following information applies to all EDLIN commands:

- The prompt is an asterisk (*).
- With the exception of the edit line command, all commands are a single letter.
- With the exception of the **E** (end edit) and **Q** (quit edit) commands, commands are usually preceded and/or followed by parameters.
- Enter commands and string parameters in uppercase, or lowercase, or a combination of both.
- Separate commands and parameters with delimiters for readability; however, a delimiter is only required between two adjacent line numbers. Remember, delimiters are spaces or commas.
- Commands become effective only after you press the Enter key.
- Stop commands by pressing the Ctrl+Break keys.
- For commands producing a large amount of output, press the Ctrl+Num Lock keys to suspend the display so that you can read it before it scrolls away. Press any other character to restart the display.
- Use the control keys and OS/2 editing keys, described in the *User's Guide*. They are very useful for editing *within a line*, while the EDLIN commands can be used for editing operations on *entire lines*.
- It is possible to refer to line numbers relative to the current line. Use a minus (-) sign and a number to indicate a line before the current line. Use a plus (+) sign and a number to indicate a line after the current line. For example:

  ```
  -10,+10L
  ```

  This command displays 21 lines: 10 lines before the current line, the current line, and 10 lines after the current line.

- Multiple commands can be entered on one command line. When you enter the command to edit a single line using [*line*], you must use a semicolon to separate the commands on the line. In the case of the **S** (search text) or **R** (replace text) command, the [*string*] can be ended by Ctrl+Z (F6) instead of the Enter key. Otherwise, one command can follow another without any special delimiting characters. For example:

  ```
  15;-5,+5L
  ```

  edits line 15 and then displays lines 10 through 20 on the screen.

- Control characters can be inserted into the text, or can be used in the strings for the **S** (search text) and **R** (replace text) commands. To enter a control character, press Ctrl+V, then enter the desired control character in uppercase. For example, the sequence Ctrl+V followed by Z generates the control character Ctrl+Z.

A (append lines)

Purpose

Adds the specified number of lines from disk to the file being edited in storage. The lines are added at the end of the current lines in storage.

Syntax



Remarks

This command is only meaningful if the file being edited is too large to fit in storage. As many lines as possible are read into storage for editing when you start EDLIN. If you do not specify the number of lines, lines are appended to storage until available storage in DOS mode is 75% full, or 64K lines have been read (whichever is smaller). No action is taken if storage is already 75% full.

To edit the remainder of the file that will not fit into storage, you must write edited lines in storage to disk by using the **A** (append lines) command. Refer to the **W** (write lines) command for information on how to write edited lines to disk.

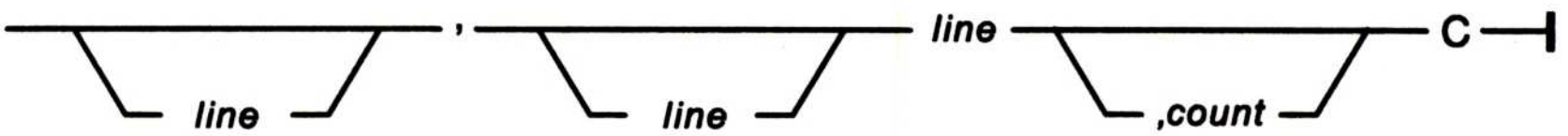
A message is displayed when the **A** (append lines) command has read the last line of the file into storage.

C (copy lines)

Purpose

Copies the lines in the specified range to the line number specified by the third parameter. The new data is placed ahead of the line that was specified in the third parameter.

Syntax



Parameters

count The operation is repeated the number of times specified in *count*.

Remarks

The parameter *count* defaults to 1. To repeat text, specify the number of times the operation is to be performed in *count*. If the first parameter or the second parameter is omitted, the default is the current line. This effectively copies the current line to the specified line. The file is renumbered accordingly. The first of the copied lines becomes the current line. For example:

```
1,5,8C
```

copies lines 1 through 5 to line 8. Line 8 becomes the current line.

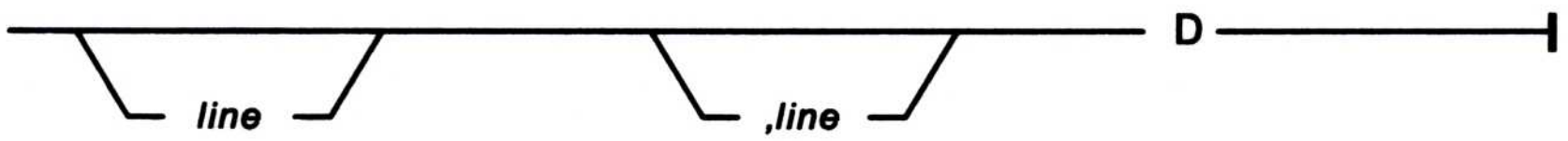
The line numbers must not overlap or an error is reported. Also, the characters - and + are not allowed in the *count* field.

D (delete lines)

Purpose

Deletes a specified range of lines.

Syntax



Remarks

The line following the deleted range becomes the current line, even if the deleted range includes the last line in storage. The current line and any following lines are renumbered.

Example

If you omit the first parameter, as in:

```
,lineD
```

deletion starts with the current line and ends with the line specified by the second parameter. The beginning comma is required to indicate the omitted first parameter.

If you omit the second parameter, as in:

```
lineD      or   line,D
```

only the one specified line is deleted. If you omit both parameters, as in:

```
D
```

only the current line is deleted, and the line that follows becomes the current line.

Assume that you want to edit the following file. The current line is line 22.

```
 1: This is a sample file used to demonstrate
 2: line deletion and dynamic
 3: line number generation.
 .              .
 .              .
 .              .
20: See what happens to the lines
21: and line numbers when lines are
22:*deleted.
23: See how easy this is to use.
```

If you want to delete a range of lines, from 3—20, enter:

```
3,20D
```

The result is:

```
1: This is a sample file used to demonstrate
2: line deletion and dynamic
3:*and line numbers when lines are
4: deleted.
5: See how easy this is to use.
```

Lines 3—20 are deleted from the file. Lines 21—23 are renumbered to 3—5. Line 3 becomes the current line. If you want to delete the current and the following line, enter:

```
,4D
```

The result is:

```
1: This is a sample file used to demonstrate
2: line deletion and dynamic
3:*See how easy this is to use.
```

Lines 3 and 4 are deleted from the file. Line 5 is renumbered to 3. Line 3 is still the current line, but now it has different text.

If you want to delete a single line, such as line 2, enter:

```
2D
```

The result is:

```
1: This is a sample file used to demonstrate
2:*See how easy this is to use.
```

D (delete lines)

Line 2 is deleted. Line 3 is renumbered to 2. The new line 2 becomes the current line. If you want to delete only the current line, enter:

```
D
```

The result is:

```
1:*This is a sample file used to demonstrate
```

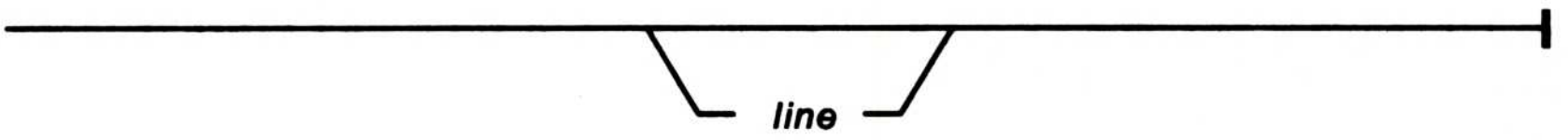
The current line, line 2, is deleted. The new line 1 becomes the current line.

Edit Line

Purpose

Allows you to edit a line of text. You must enter the line number of the line to be edited, or enter a period (.) to indicate the current line.

Syntax



Remarks

If you just press Enter, you specify that the line after the current line is to be edited. The line number and its text are displayed and the line number is repeated on the line below. When you press the Enter key, the edited line is placed in the file and becomes the current line.

If you decide not to save the changed line, press either Esc or Ctrl+Break instead of Enter. The original line remains unchanged. Pressing the Enter key with the cursor at the beginning of the line has the same effect as pressing Esc or Ctrl+Break.

If the cursor is in any position other than the beginning or the end of a line, pressing Enter erases the rest of the line.

Example

Assume that you want to edit line 6 and you have entered its number. The following is displayed on the screen:

```
*6
   6: This is a sample unedited line.
   6: _
```

The first line is your request to edit line 6, followed by the two-line display response.

If you want to move the cursor to the letter **u**, press F2 and enter:

```
u
```

The result is:

```
*6

   6: This is a sample unedited line.
   6: This is a sample _
```

If you want to delete the next two characters and keep the remainder of the line, press Del twice; then press F3.

The result is:

```
*6
   6: This is a sample unedited line.
   6: This is a sample edited line._
```

Now you can take one of the following actions:

- Press Enter to save the changed line.
- Extend the changed line by typing more text. You are automatically in insert mode when the cursor is at the end of a line.
- Press F5 to do additional editing to the changed line without changing the original line.
- Press Esc or Ctrl+Break to cancel the changes you made to the line. The original contents of the line are preserved.

E (end edit)

Purpose

Ends EDLIN and saves the edited file.

Syntax



Remarks

The edited file is saved by writing it to the drive and filename specified when you started EDLIN. The original file, the one specified when EDLIN was started, is given a .BAK filename extension. A .BAK file is not created if there is no original file; that is, if you created a new file instead of updating an old file during the editing session.

EDLIN returns to the OS/2 command processor, which displays the command prompt.

Be sure that your disk has enough free space to save the entire file. If your disk does not have enough free space, only a portion of the file is saved. The portion in storage that is not written to disk is lost. In this case, your original file is not renamed to .BAK, and the portion of data that was written to disk will have a filename extension of $$$.

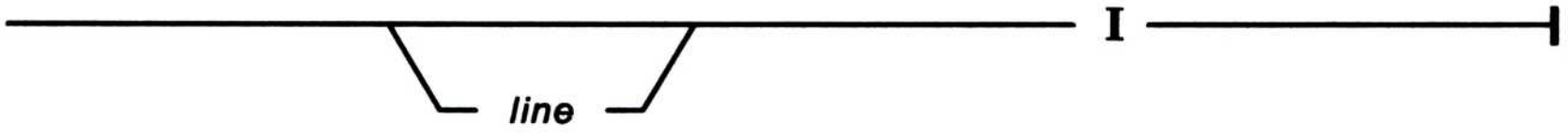
EDLIN appends a carriage return, line feed sequence to the end of the file, if it was not already present, to delimit the last line of text in the file. Also, a Ctrl+Z character is added as the last character in the saved file. This serves as an end-of-file mark.

I (insert lines)

Purpose

Inserts lines of text immediately *before* the specified line. When you create a new file, you must enter the **I** (insert lines) command before text can be inserted.

Syntax



Remarks

If you do not specify line, or if you specify line as a period (.), the insert is made immediately before the current line. If the line number you specify is greater than the highest existing line number, or if you specify # as the line number, the insertion is made after the last line in storage.

EDLIN displays the appropriate line number so that you can enter more lines, ending each line by pressing Enter. During the insert mode of operation, successive line numbers appear automatically each time Enter is pressed. You must press Ctrl+Break to discontinue the insert mode of operation.

The line that follows the inserted lines becomes the current line, even if the inserted lines are added to the end of the lines in storage. The current line and any remaining lines are renumbered.

Example

Assume that you want to edit the following file. Line 2 is the current line.

```
1: This is a sample file used to demonstrate
2:*line deletion
3: and dynamic line number generation.
```

If you want to insert text before line 3, the entry and immediate response look like this:

```
*3I
   3:*_
```

Now, if you want to insert two new lines of text, enter:

```
*3 I
   3:*First new line of text
   4:*Second new line of text
   5:*
```

and press Ctrl+Break.

The original line 3 is now renumbered to line 5.

If you display the file with a **L** (list lines) command, the file looks like this:

```
1: This is a sample file used to demonstrate
2: line deletion
3: First new line of text
4: Second new line of text
5:*and dynamic line number generation.
```

If the two lines that were inserted were placed at the beginning of the file, the screen would look like this:

```
1: First new line of text
2: Second new line of text
3:*This is a sample file used to demonstrate
4: line deletion
5: and dynamic line number generation.
```

If the two lines that were inserted were placed immediately before the current line (2 I or . I or I), the screen would look like this:

```
1: This is a sample file used to demonstrate
2: First new line of text
3: Second new line of text
4:*line deletion
5: and dynamic line number generation.
```

I (insert lines)

If the two inserted lines were placed at the end of the file (4 I or # I), the screen would look like this:

```
1: This is a sample file used to demonstrate
2: line deletion
3: and dynamic line number generation.
4: First new line of text
5: Second new line of text
```

L (list lines)

Purpose

Displays a specified range of lines. The current line remains unchanged.

Syntax



Remarks

Default values are provided if either one or both of the parameters are omitted. If you omit the first parameter, as in:

```
,lineL
```

the display starts 11 lines before the current line and ends with the specified line. The beginning comma is required to indicate the omitted first parameter.

Note: If the specified line is more than 11 lines before the current line, the display is the same as if you omitted both parameters.

If you omit the second parameter, as in:

```
lineL      or   line,L
```

starting with the specified line, 23 lines are displayed.

If you omit both parameters, as in:

```
L
```

the 11 lines before the current line, the current line, and the 11 lines after the current line are displayed. If there aren't 11 lines before the current line, then extra lines are displayed after the current line to make a total of 23 lines.

L (list lines)

Example

Assume that you want to edit the following file. Line 15 is the current line.

```
 1: This is a sample file used to demonstrate
 2: line deletion and dynamic
 3: line number generation.
 .              .
 .              .
 .              .
15:*This is the current line (note the asterisk)
 .              .
 .              .
 .              .
25: See what happens to the lines
26: and line numbers when lines are
27: deleted.
```

If you want to display a range of lines, from 3—25, enter:

```
3,25L
```

The screen looks like this:

```
 3: line number generation.
 .              .
 .              .
 .              .
15:*This is the current line (note the asterisk)
 .              .
 .              .
 .              .
25: See what happens to the lines
```

If you want to display the first three lines, enter:

```
1,3L
```

The screen looks like this:

```
 1: This is a sample file used to demonstrate
 2: line deletion and dynamic
 3: line number generation.
```

If you want to display 23 lines of the file, starting with line 3, enter:

```
3L
```

The screen looks like this:

```
 3: line number generation.
  .              .
  .              .
  .              .
15:*This is the current line (note the asterisk)
  .              .
  .              .
  .              .
25: See what happens to the lines
```

If you want to display 23 lines centered around the current line, enter:

```
L
```

The screen looks like this:

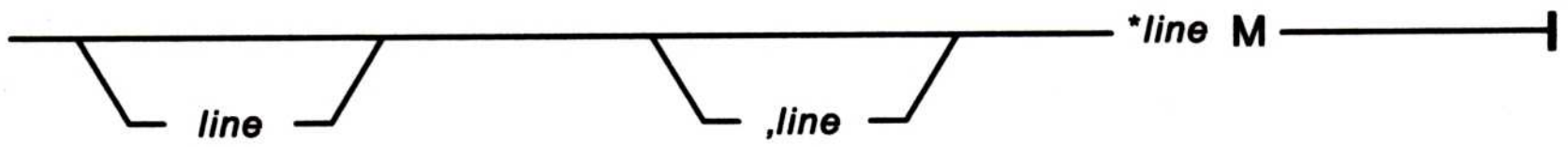
```
 4: Fourth line of text
 5: Fifth line of text
  .              .
  .              .
  .              .
15:*This is the current line (note the asterisk)
  .              .
  .              .
  .              .
25: See what happens to the lines
26: and line numbers when lines are
```

M (move lines)

Purpose

Moves the range of lines specified by the first two *line* parameters ahead of the line specified in the third *line* parameter. The third parameter is not optional.

Syntax



Remarks

Use this command to move a block of data from one location in the file to another. If the first or second line parameter is omitted, it will default to the current line. After the move, the first of the moved lines becomes the current line. The lines are renumbered according to the direction of the move. For example:

```
,+25,100M
```

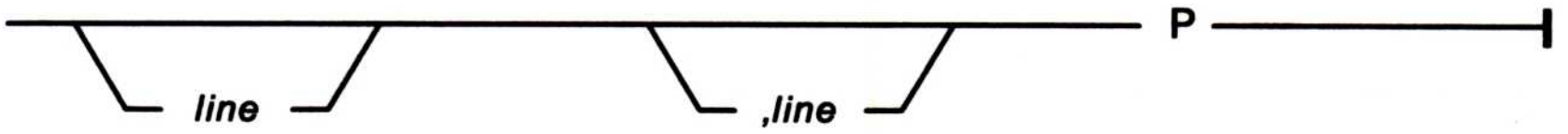
moves the data from the current line plus 25 lines to line 100. If the line parameters overlap, an entry error is reported.

P (page)

Purpose

Lists the specified block of lines.

Syntax



Remarks

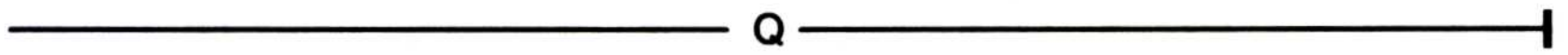
If the first *line* parameter is omitted, it defaults to the current line plus one. If the second *line* parameter is omitted, 23 lines are listed. The new current line becomes the last line displayed by the **P** (page) command and is marked with an asterisk. This command pages through a file displaying 23 lines at a time. It differs from the **L** (list lines) command in that it changes the current line.

Q (quit edit)

Purpose

Quits the editing session without saving any changes you may have entered.

Syntax



Remarks

EDLIN prompts you to make sure you really don't want to save the changes.

Enter **Y** if you want to quit the editing session. No editing changes are saved and no .BAK file is created. Refer to the **E** (end edit) command for information about the .BAK file.

Enter **N**, or any other character, if you want to continue the editing session.

Example

When you enter the **Q** command, the following is displayed:

```
Q
Do you want to end the edit (Y/N)?_
```

R (replace text)

Purpose

Replaces all occurrences of the first string in the specified range of lines with the second string.

Notes:

1. If you omit the second string, **R** (replace text) deletes all occurrences of the first string within the specified range of lines. If you omit both strings, EDLIN reuses the search string entered with the most recent **S** or **R** command, and the replacement string entered with the last **R** command.
2. This command uses the F6 key as normally set up by OS/2. If you have changed the meaning of the F6 key through "Extended Keyboard Control," use the keys Ctrl+Z where F6 is referred to below.

EDLIN displays the changed lines each time they are changed. The last line changed becomes the current line.

Syntax



Parameters

? You can specify the optional parameter ? to request a prompt (**O.K.?**) after each display of a modified line. Press Y or the Enter key if you want to keep the modification.

Enter any other character if you don't want the modification. In either case, the search continues for further occurrences of the first string within the range of lines, including multiple occurrences within the same line.

F6 The first string begins with the character in the position immediately following R, and continues until you press F6 or Ctrl+Z (or the Enter key if the second string is omitted).

R (replace text)

The second string begins immediately after you press F6 or Ctrl+Z and continues until you press Enter.

Remarks

Defaults occur if either one or both of the *line* parameters are missing.

If you omit the first *line*, the search begins with the line after the current line. If you omit the second *line*, the search ends with the last line in storage. If you omit both *line* parameters, the system searches from the line following the current line to the last line in storage. Strings that are identical to the first string are looked for; that is, if the first string is a combination of uppercase and lowercase letters, only strings with identical characters are replaced.

Example

Assume that you want to edit the following file. Line 7 is the current line.

```
1: This is a sample file
2: used to demonstrate
3: the R and S commands.
4: This includes the
5: optional parameter ?
6: and required string
7:*parameter.
```

To replace all occurrences of *and* with *or* in the lines in storage, enter:

```
1,7 Rand
```

Then press F6, type **or**, and press Enter.

The result is:

```
3: the R or S commands.
3:*the R or S commors.
6: or required string
```

Line 6 becomes the current line in the file because line 6 was the last line changed. Notice that lines 1, 2, 4, 5, and 7 are not displayed because they were not changed.

Greater selectivity can be achieved by requesting a prompt (by using the ? parameter) after each display of a modified line. If you request a prompt, the screen looks like this:

```
*1,7 ? Rand
     3: the R or S commands
O.K.? Y
     3: the R or S commors
O.K.? N
     6: or required string
O.K.? Y
*
```

Lines 3 and 6 are displayed like this:

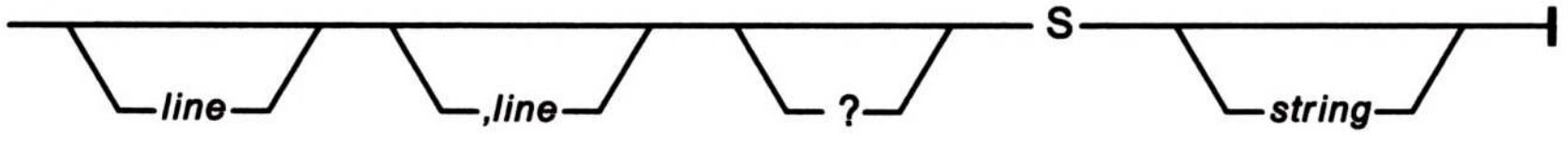
```
3: the R or S commands.
6: or required string
```

S (search text)

Purpose

Searches a specified range of lines to locate a specified string.

Syntax



Parameters

?	You should specify the optional parameter ? if you would like a prompt (**O.K.?**) after each display of a line containing the specified string.

Remarks

The first line to contain the specified string is displayed and the search ends (unless you use the ? parameter). The first line found that contains the specified string becomes the current line.

Note: The **S** (search text) command always searches for the identical character in text. That is, it searches for uppercase if you enter uppercase, and lowercase if you enter lowercase.

If you do not enter a string, the **S** command uses the last search string entered on a **R** (replace text) or **S** command. If the specified string is not found, the search ends and the message **Not found** is displayed. The current line remains unchanged. If you enter **Y** or press the Enter key, the line that matches the specified string becomes the current line and the search ends. Enter any other character to continue the search until another string is found, or until all lines within the range are searched. Once all the lines within the range are searched, the **Not found** message is displayed.

The system provides default values if you omit the first, second, or both line parameters. If you omit the first line parameter, the system defaults to the line following the current line. If you omit the second line parameter, the system defaults to the last line in storage. If you

omit both line parameters, the system searches from the line following the current line to the last line in storage.

Notes:

1. The string begins with the character in the position immediately following the **S** and continues until you end the string by pressing the Enter key.
2. If you wish to place more than one command on a line containing an **S** command, the **S** command should end in a Ctrl+Z (F6), and the next command should begin in the following character position.

Example

Assume that you want to edit the following file. Line 7 is the current line.

```
1: This is a sample file
2: used to demonstrate
3: the S command.
4: This includes the
5: optional parameter ?
6: and required string
7:*parameter.
```

If you want to search for the first occurrence of **and** in the file, enter:

```
1,7 Sand
   or
1, Sand
   or
1Sand
```

The result is:

```
     3: the S command.
*
```

The *and* is part of the word *command*. Notice that line 3 becomes the current line in the file.

Perhaps this is not the *and* you were looking for. To continue the search, simply reenter **S** and press Enter. The search continues with the line following the current line (the line just found).

S (search text)

The screen looks like this:

```
*1,7 Sand
     3: the S command.
*S
     6: and required string
*
```

Line 6 now becomes the current line in the file.

You can also search for strings by requesting a prompt (by means of the ? parameter) after each display of a matching line. In this case, the screen looks like this:

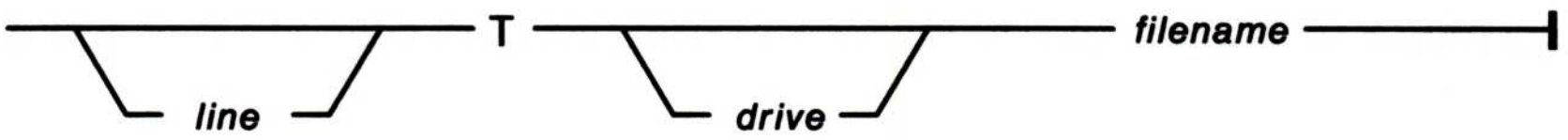
```
*1,7 ? Sand
     3: the S command.
O.K.? N
     6: and required string
O.K.? Y
*
```

T (transfer lines)

Purpose

Transfers (merges) the contents of a specified file into the file currently being edited.

Syntax



Remarks

The *filename* contents are merged ahead of the *line* in the file being edited. If *line* is omitted, the current line is used.

Note: The file being merged is read from the current directory of the specified or default drive. If a path was specified when you issued the EDLIN command, that path becomes the current directory for that drive for the duration of the EDLIN session, and any **T** (transfer lines) commands for that drive must be satisfied from the same directory.

W (write lines)

Purpose

Writes a specified number of lines to disk from the lines being edited in storage. Lines are written beginning with line number 1.

Syntax



Remarks

This command is only meaningful if the file you are editing is too large to fit in storage. When you start EDLIN, it reads lines into storage until available storage in DOS mode is 75% full, or until approximately 64K characters have been read (whichever amount is smaller).

To edit the remainder of the file not in storage, you must write edited lines in storage to disk. Then you can load additional unedited lines from disk into storage by using the **A** (append lines) command.

Note: If you do not specify the number of lines, lines are written until 25% of available DOS mode storage is used, or until 16K characters remain in storage (whichever amount is smaller). No action is taken if available storage is already less than 25% used. All lines are renumbered so that the first remaining line becomes number 1.

Appendix D. Keyboard Templates

You can use the keyboard templates in this appendix to locate character positions on the keyboard after you have changed your keyboard layout with the KEYB command. Function keys located across the top of the keyboard are not shown. Shaded areas indicate that keys are not available on some U.S. keyboards.

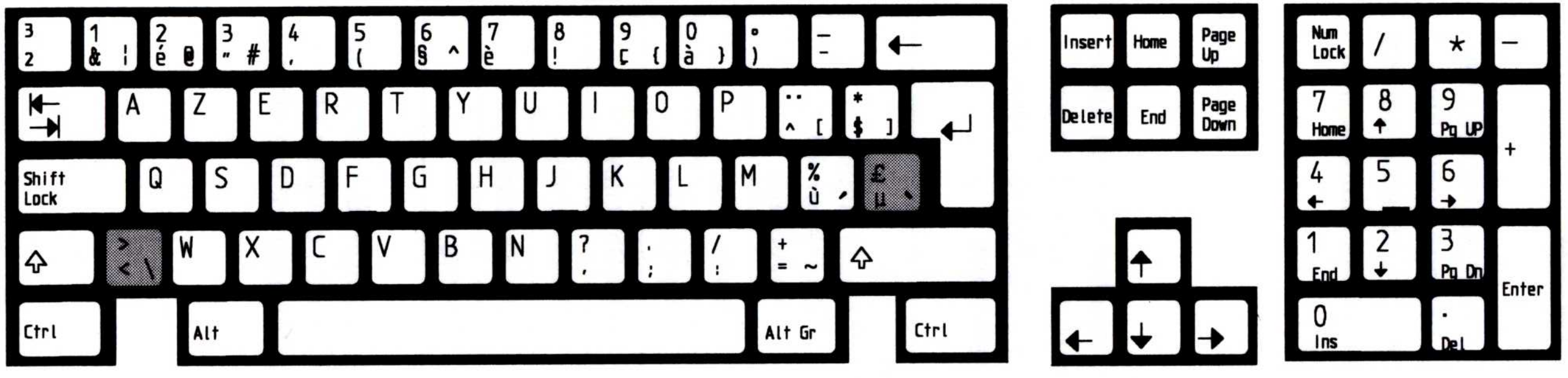
Multiple Character Keys

For keys with three or four characters shown, use the key combinations in the following table to produce the desired upper right and lower right characters. No additional keys are required for lower left characters. Use the shift key for upper left characters.

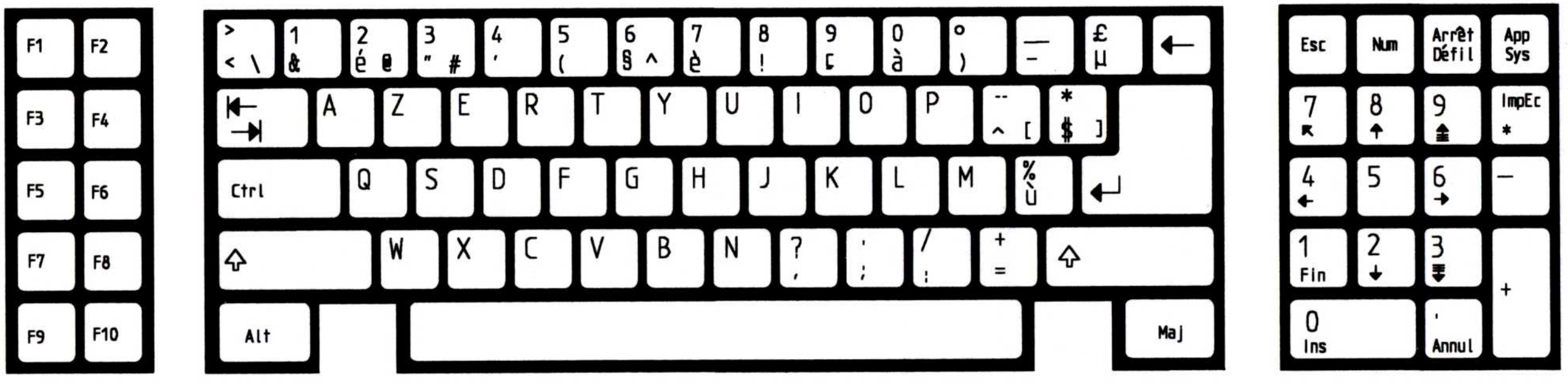
Country	Lower Right Character Except Enhanced Keyboard	Lower Right Character Enhanced Keyboard	Upper Right Character All Keyboards
Canada (Fr.)	Alt-Shift keys	Alt-Gr.	None
Denmark	Alt	Alt-Gr.	Alt-Shift keys
Finland	Alt	Alt-Gr.	Alt-Shift keys
Norway	Alt	Alt-Gr.	Alt-Shift keys
Sweden	Alt	Alt-Gr.	Alt-Shift keys
All others	Alt-Ctrl keys	Alt-Gr.	None

Refer to Appendix E, "Accented Characters" on page E-1 for allowable accent key and letter key combinations for each country keyboard and code page combination. For an explanation of the KEYB command, see page 2-81.

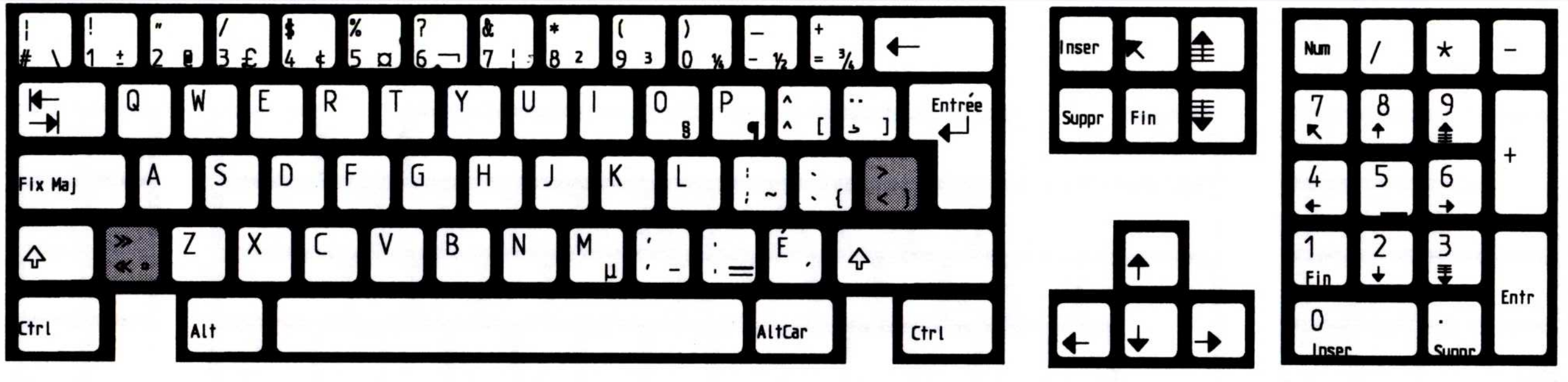
Belgium Enhanced PC



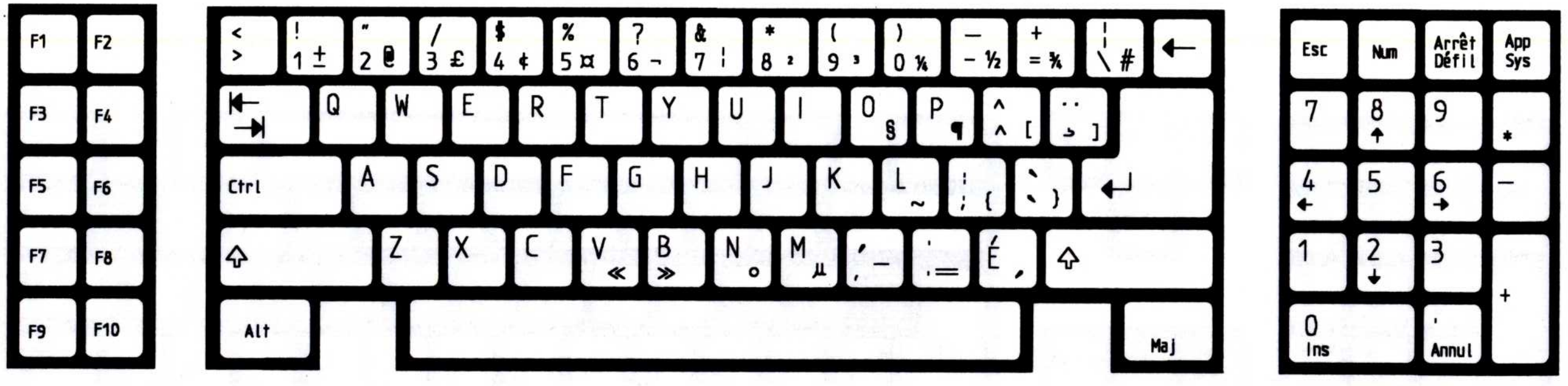
Belgium AT



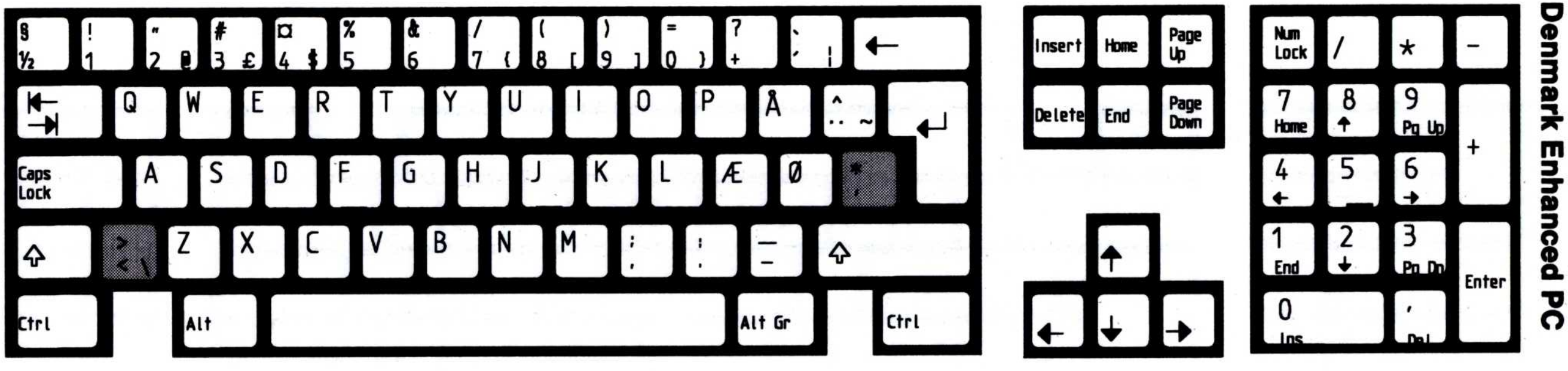
Canada Enhanced PC



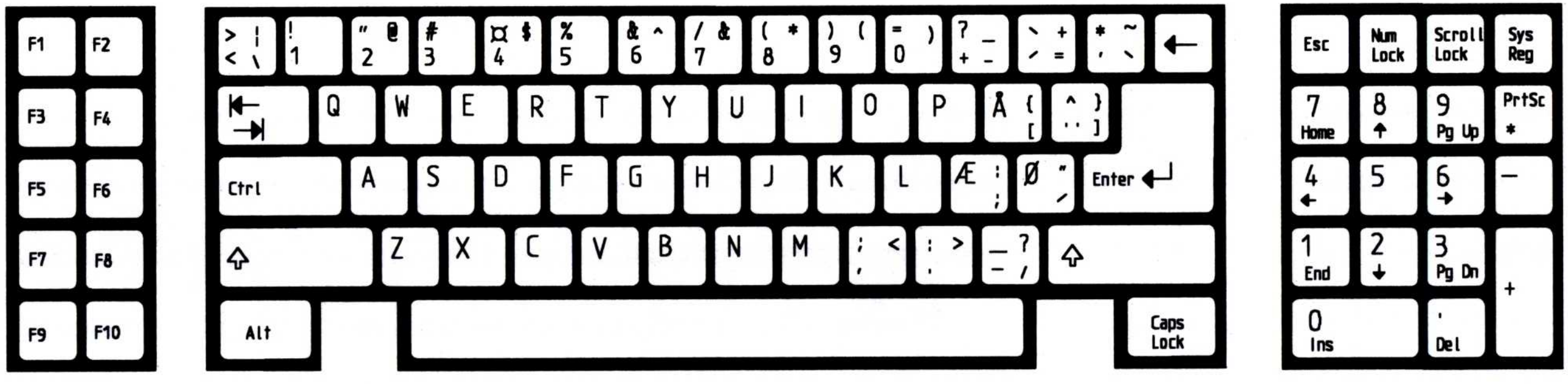
Canada AT



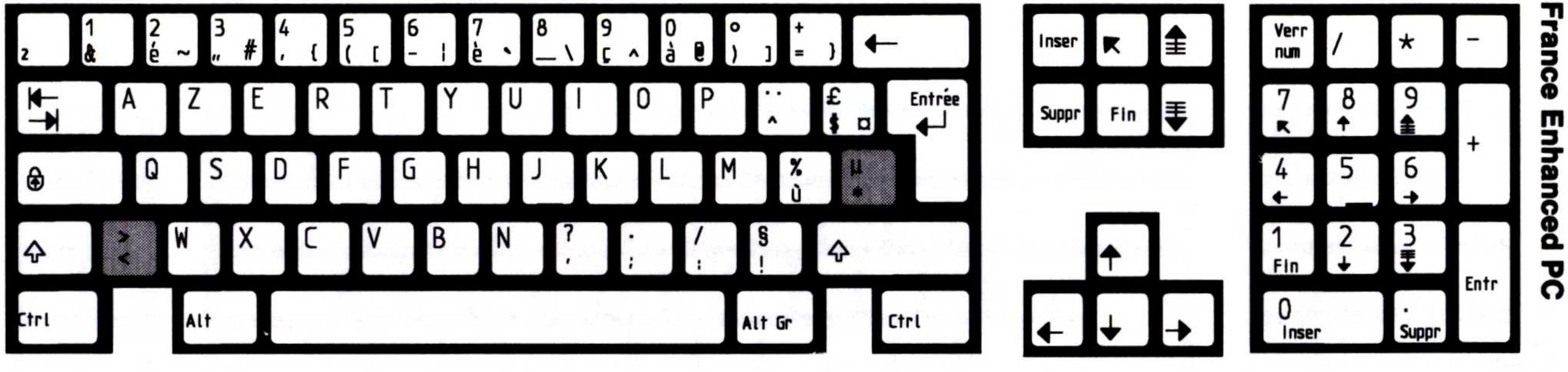
Denmark Enhanced PC



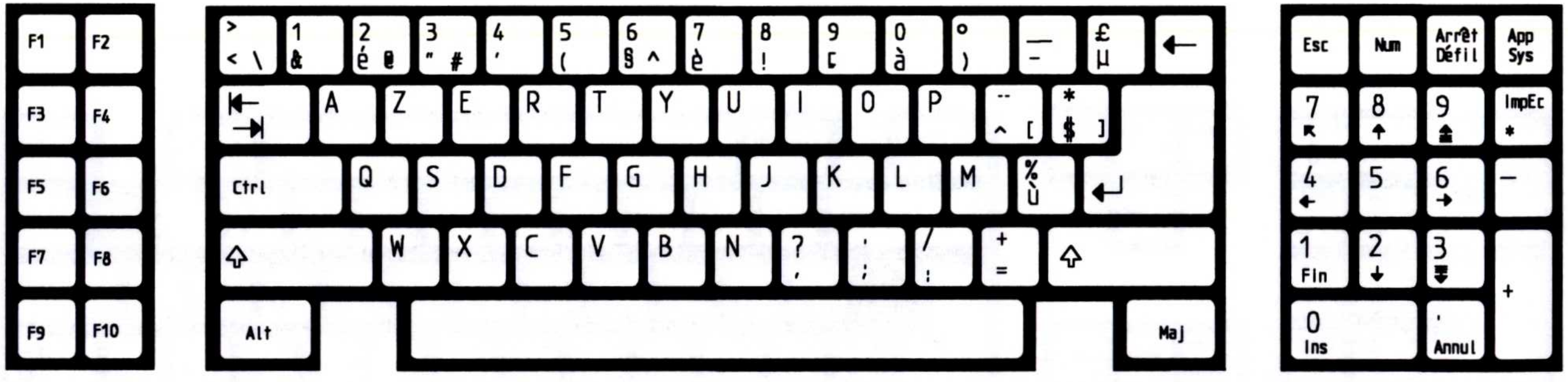
Denmark AT



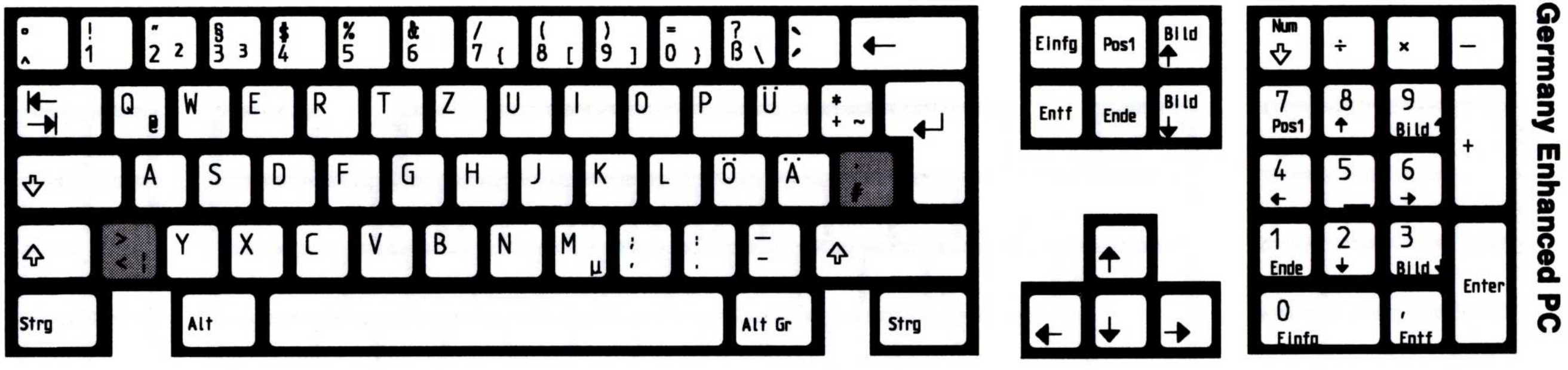
France Enhanced PC



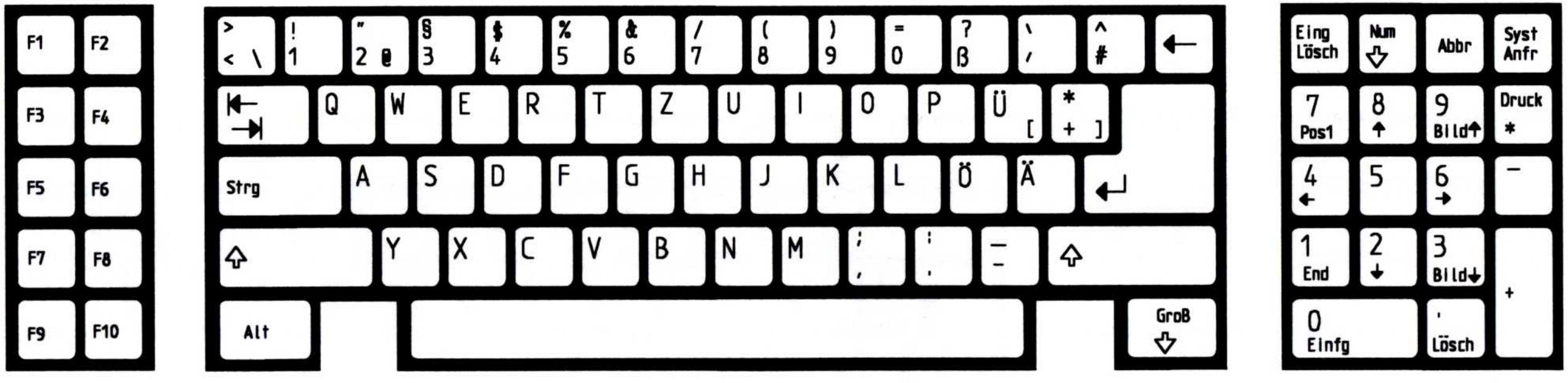
France AT



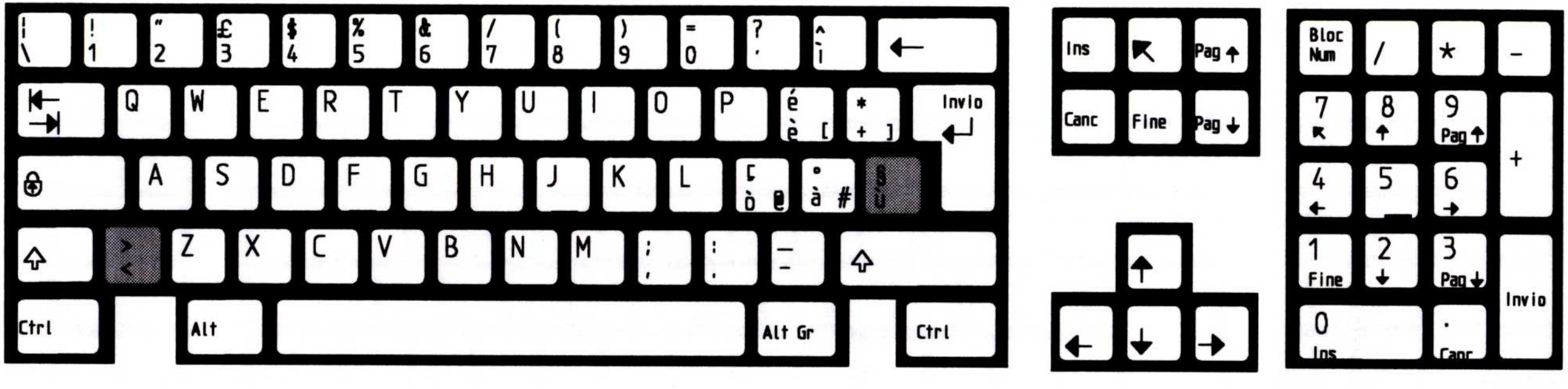
Germany Enhanced PC



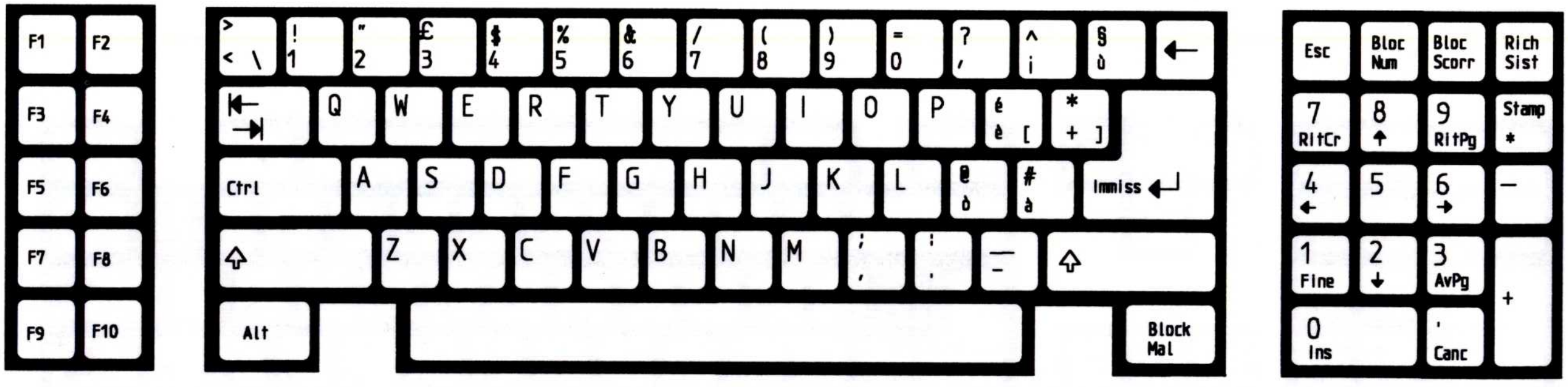
Germany AT



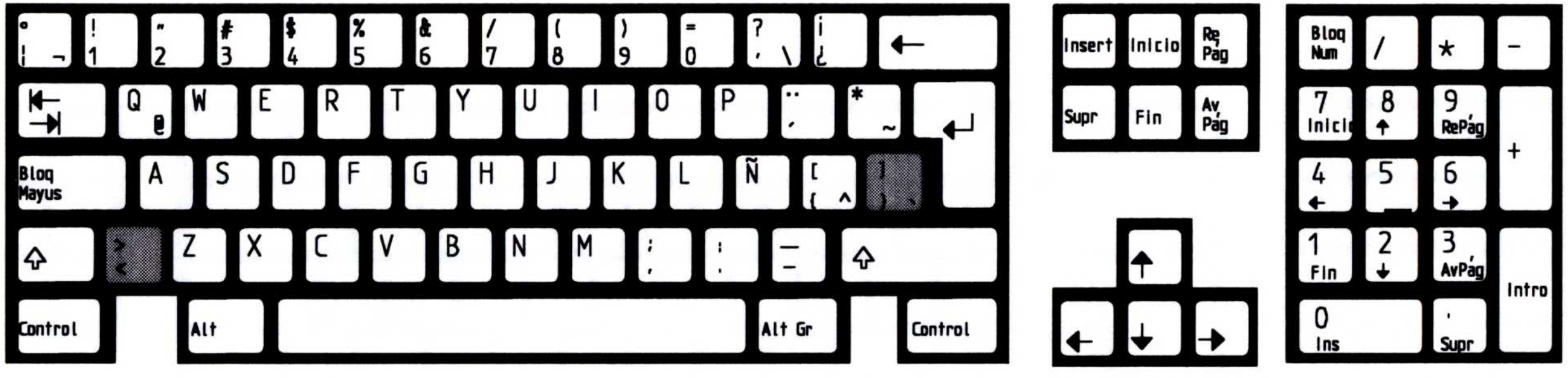
Italy Enhanced PC



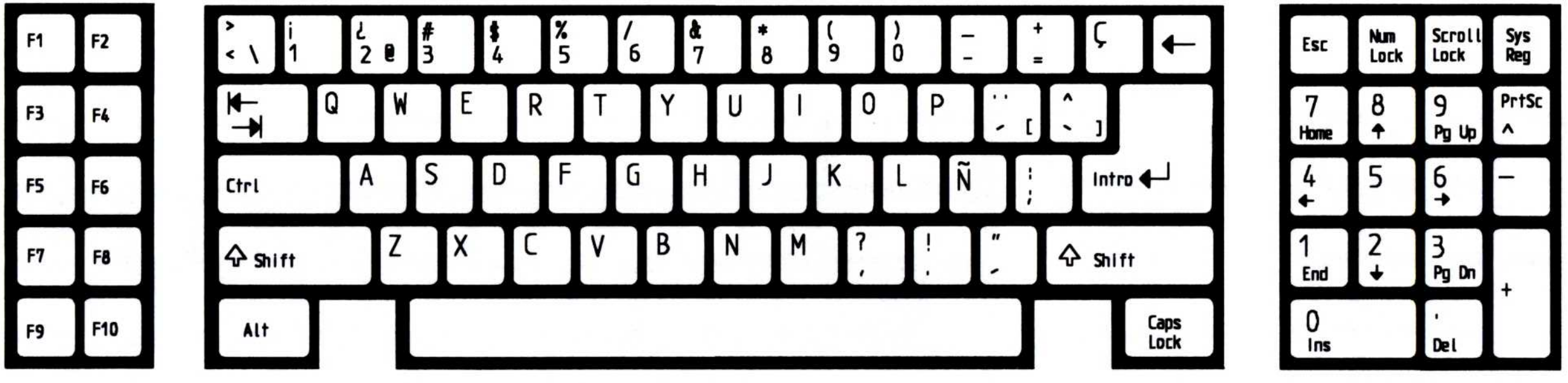
Italy AT



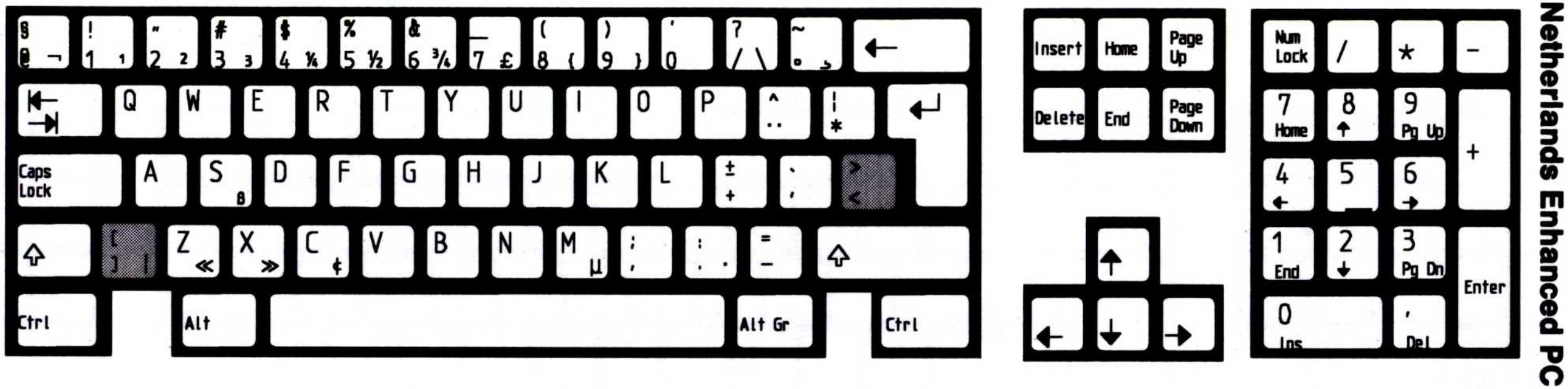
Latin America Enhanced PC



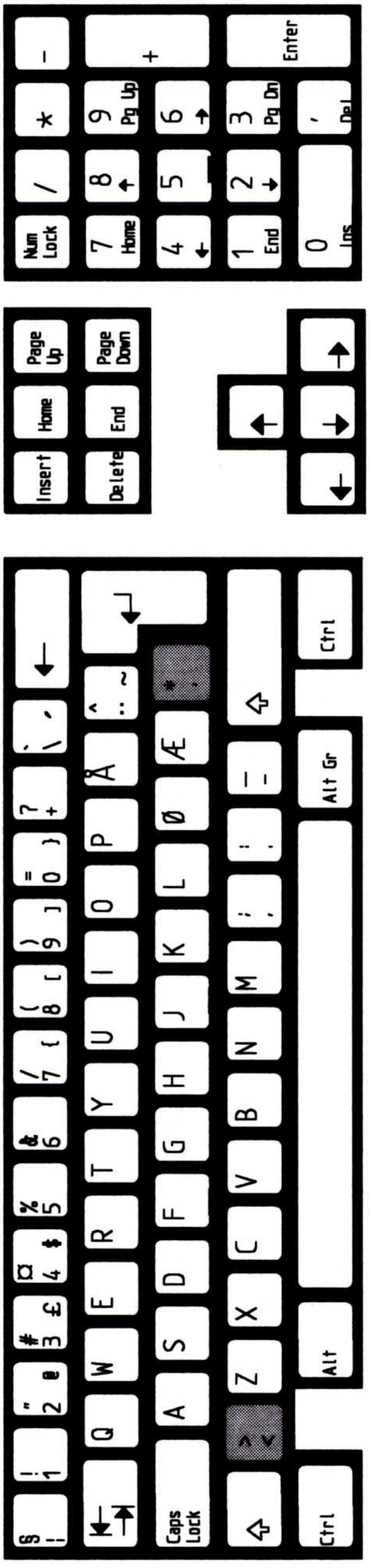
Latin America AT



Netherlands Enhanced PC



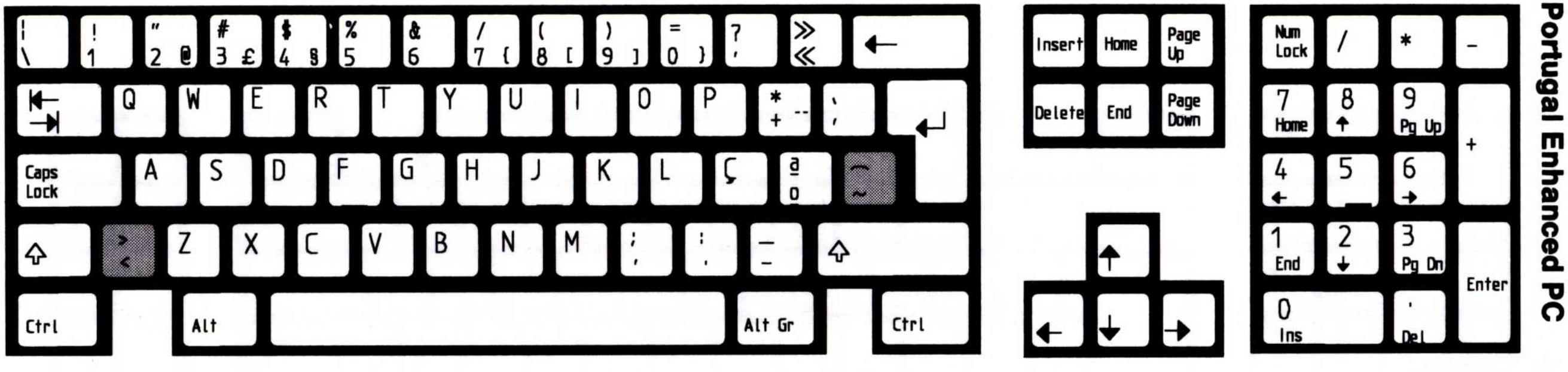
Norway Enhanced PC



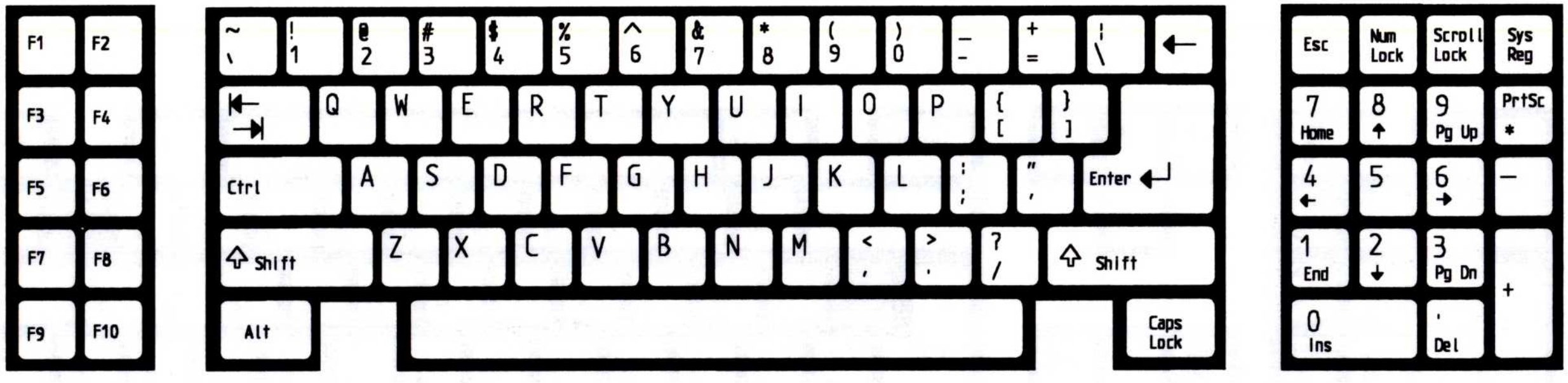
Norway AT



Portugal Enhanced PC



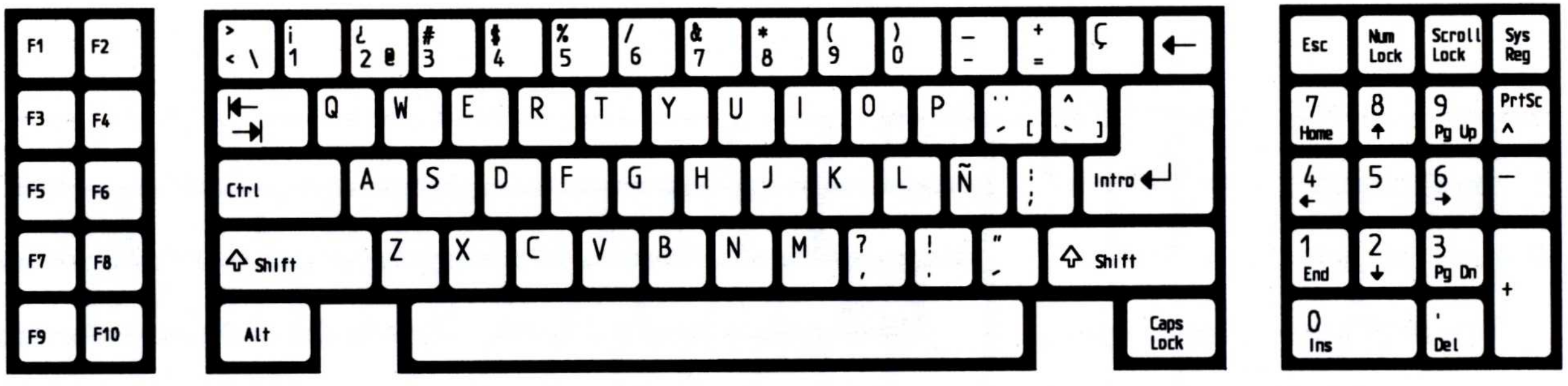
Portugal AT



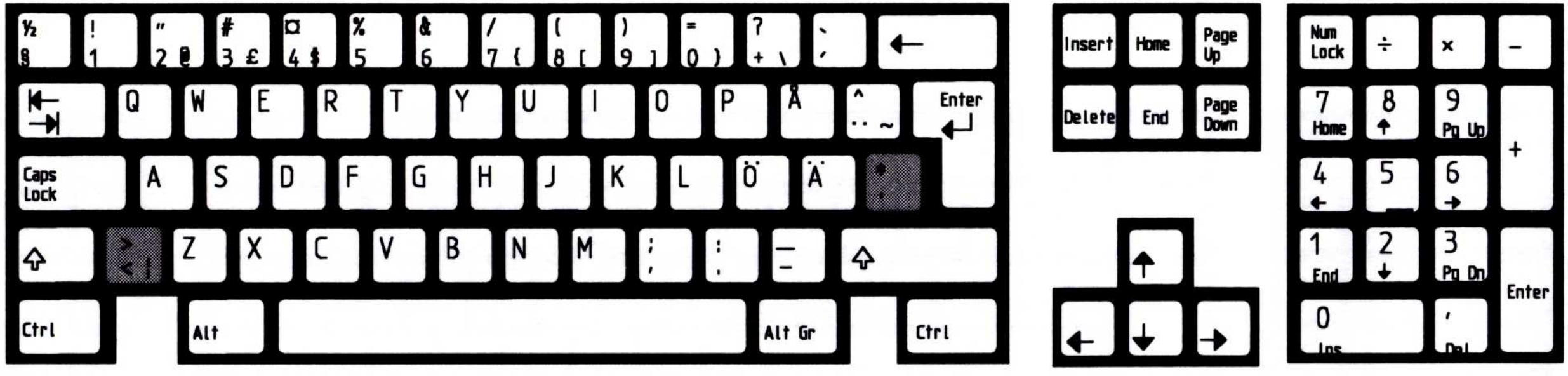
Spain Enhanced PC



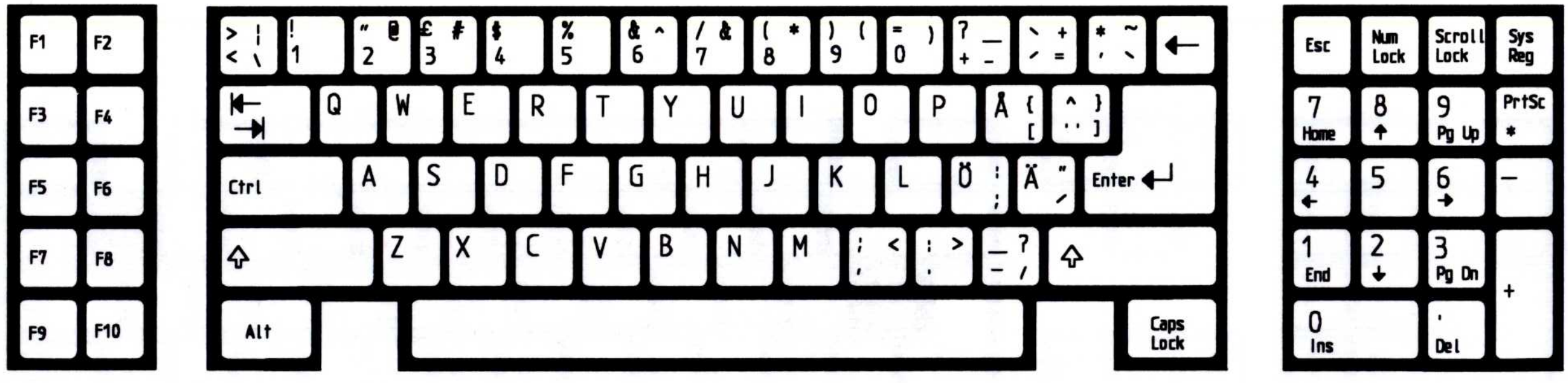
Spain AT



Sweden/Finland Enhanced PC



Sweden/Finland AT



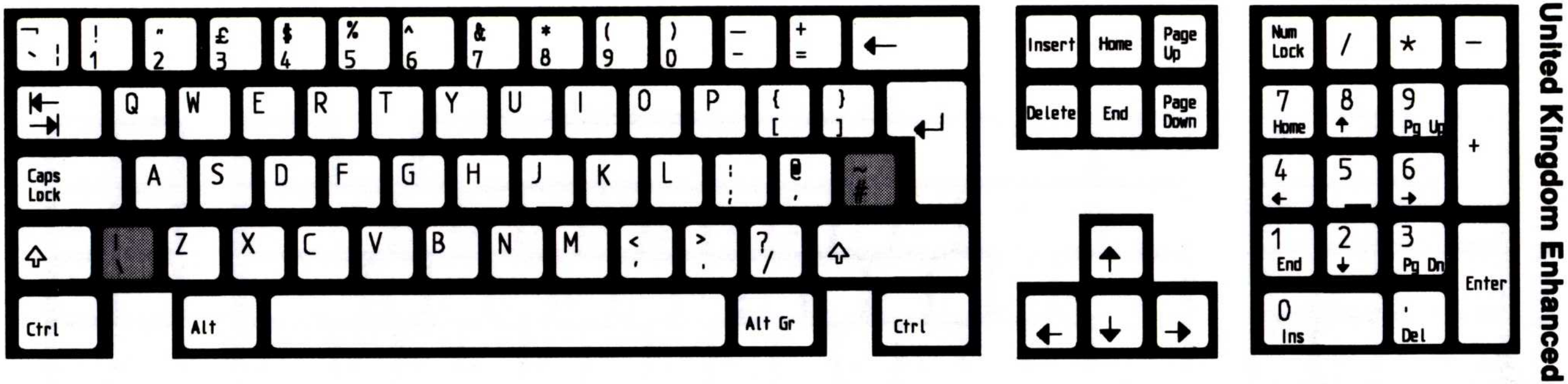
Switzerland Enhanced PC



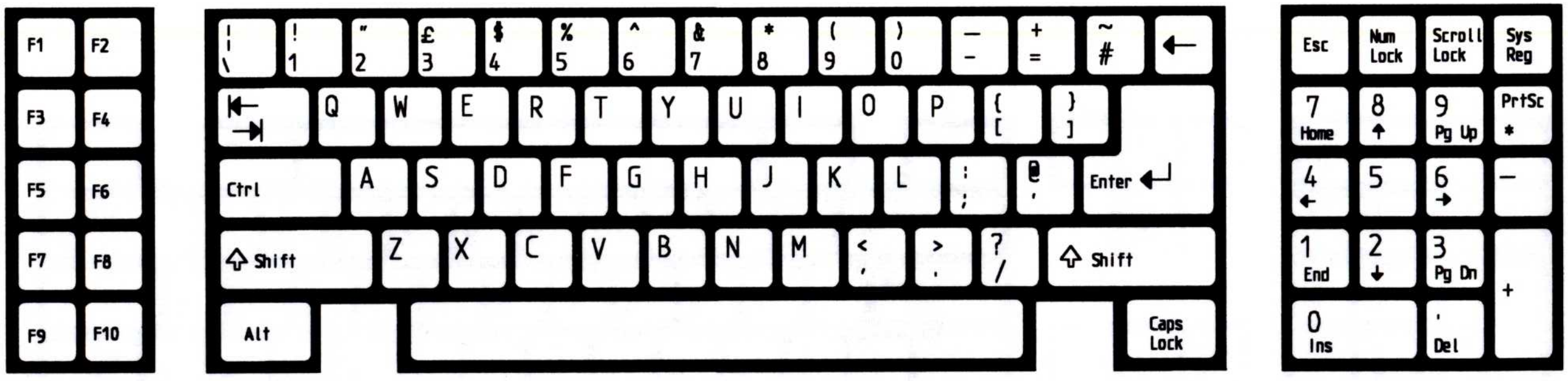
Swizerland AT

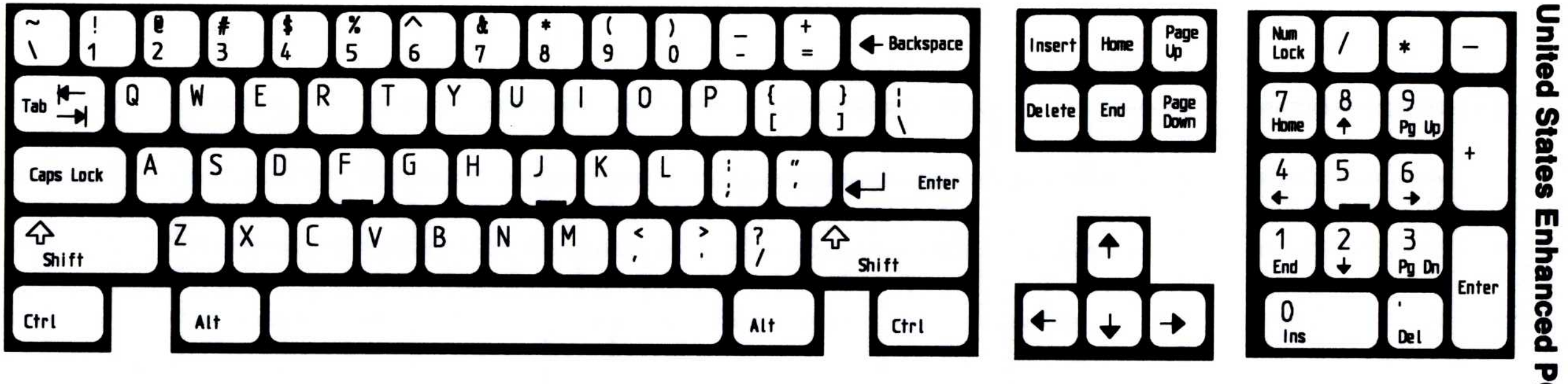


United Kingdom Enhanced PC

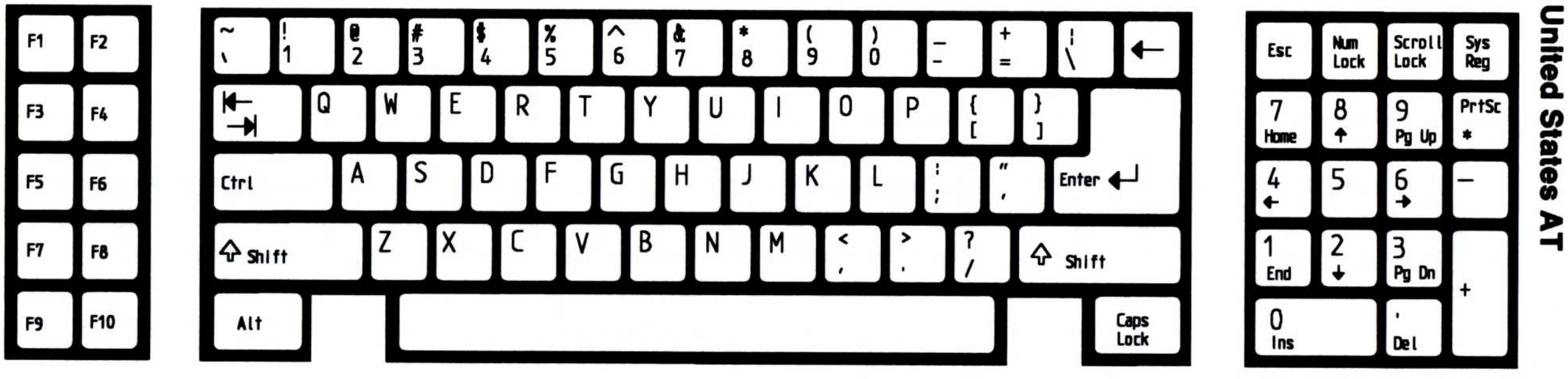


United Kingdom AT





United States Enhanced PC



United States AT

Appendix E. Accented Characters

This appendix shows you the accented characters obtainable with each keyboard and code page combination for the countries supported by Operating System/2™ (OS/2™).

Nonescaping Key Combinations

Allowable nonescaping key combinations are listed by country on the following page. Italy, the United Kingdom, and the United States do not have accented characters.

France (AT and Enhanced Keyboards)
437: äÄëïöÖüÜÿ âêîôû
850: äÄëËïÏöÖüÜÿ âÂîÎôÔûÛ
Germany (AT and Enhanced Keyboards)
437: ääÉíóú àèìòù
850: áÁéÉíÍóÓúÚ àÀèÈìÌòÒùÙ
Spain (AT and Enhanced Keyboards)
437: ääÉíóú àèìòù äÄëïöÖüÜÿ âêîôû
850: áÁéÉíÍóÓúÚ àÀèÈìÌòÒùÙ äÄëËïÏöÖüÜÿ âÂêÊîÎôÔûÛ
Belgium (AT Keyboard: see France)
Belgium (Enhanced Keyboard)
437: ääÉíóú àèìòù äÄëïöÖüÜÿ âêîôû ñÑ
850: áÁéÉíÍóÓúÚ àÀèÈìÌòÒùÙ äÄëËïÏöÖüÜÿ âÂêÊîÎôÔûÛ ñÑãÃõÕ
Norway and Denmark (AT Keyboard)
865: ääÉíóú àèìòù äÄëïöÖüÜÿ âêîôû
850: áÁéÉíÍóÓúÚýÝ àÀèÈìÌòÒùÙ äÄëËïÏöÖüÜÿ âÂêÊîÎôÔûÛ
Norway and Denmark (Enhanced Keyboard)
865: Same combinations as AT Keyboard plus ñÑ
850: Same combinations as AT Keyboard plus ñÑãÃõÕ
Finland and Sweden (AT Keyboard)
437: ääÉíóú àèìòù äÄëïöÖüÜÿ âêîôû
850: áÁéÉíÍóÓúÚýÝ àÀèÈìÌòÒùÙ äÄëËïÏöÖüÜÿ âÂêÊîÎôÔûÛ
Finland and Sweden (Enhanced Keyboard)
437: Same combinations as AT Keyboard plus ñÑ
850: Same combinations as AT Keyboard plus ñÑ
Portugal (AT Keyboard: Uses U.S. AT Keyboard)
Portugal (Enhanced Keyboard)
860: áÁéÉíÍóÓúÚ àÀèÈìÌòÒùÙ üÜ âÂêÊôÔ ãÃõÕñÑ
850: áÁéÉíÍóÓúÚýÝ àÀèÈìÌòÒùÙ äÄëËïÏöÖüÜÿ
âÂêÊîÎôÔûÛ ãÃõÕñÑ
Canadian French (AT and Enhanced Keyboards)
863: ééÓú àÀèÈùÙ ëËïÏüÜ âÂêÊîÎôÔûÛ çÇ
850: áÁéÉíÍóÓúÚýÝ àÀèÈìÌòÒùÙ äÄëËïÏöÖüÜÿ
âÂêÊîÎôÔûÛ çÇ
Swiss French and Swiss German (AT and Enhanced Keyboards)
437: ääÉíóú àèìòù äÄëïöÖüÿ âêîôû ñÑ
850: áÁéÉíÍóÓúÚýÝ àÀèÈìÌòÒùÙ äÄëËïÏöÖüÜÿ ãÃõÕñÑ
Latin America (AT Keyboard: see Spain)
Latin America (Enhanced Keyboard)
437: ääÉíóú àèìòù äÄëïöÖüÜÿ âêîôû
850: áÁéÉíÍóÓúÚýÝ àÀèÈìÌòÒùÙ äÄëËïÏöÖüÜÿ
âÂêÊîÎôÔûÛ
Netherlands (AT Keyboard: Uses U.S. AT Keyboard)
Netherlands (Enhanced Keyboard)
437: ääÉíóú àèìòù äÄïöÖüÜÿ âêîôû ñÑ çÇ
850: áÁéÉíÍóÓúÚýÝ àÀèÈìÌòÒùÙ äÄëËïÏöÖüÿ
âÂêÊîÎôÔûÛ ãÃõÕñÑ çÇ

Appendix F. System Code Pages

The alphanumeric, graphic, and control characters contained in the system code pages supported by Operating System/2 are shown on the following pages.

Using Code Pages

If your system has been set up for code page switching, you can switch to an alternate code page. For an explanation of the purpose of code page switching, refer to the CHCP (change code page) command on page 2-22. For information on how to prepare your system and devices for successful code page switching, see "Code Page Switching" on page 4-3.

Note: It is recommended that both files and subdirectories be named using only the characters A-Z and 0-9. This prevents file access problems when switching between code pages and their associated character capitalization rules.

Code Page 437 (U.S. IBM PC)

Hex Digits 1st → 2nd ↓	0-	1-	2-	3-	4-	5-	6-	7-		8-	9-	A-	B-	C-	D-	E-	F-
-0		►		0	@	P	`	p		Ç	É	á	▒	└	╨	α	≡
-1	☺	◄	!	1	A	Q	a	q		ü	æ	í	▒	┴	╤	β	±
-2	☻	↕	"	2	B	R	b	r		é	Æ	ó	▒	┬	╥	Γ	≥
-3	♥	!!	#	3	C	S	c	s		â	ô	ú	│	├	╙	π	≤
-4	♦	¶	$	4	D	T	d	t		ä	ö	ñ	┤	─	╘	Σ	⌠
-5	♣	§	%	5	E	U	e	u		à	ò	Ñ	╡	┼	╒	σ	⌡
-6	♠	▬	&	6	F	V	f	v		å	û	ª	╢	╞	╓	µ	÷
-7	•	↨	'	7	G	W	g	w		ç	ù	º	╖	╟	╫	τ	≈
-8	◘	↑	(	8	H	X	h	x		ê	ÿ	¿	╕	╚	╪	Φ	°
-9	○	↓	)	9	I	Y	i	y		ë	Ö	⌐	╣	╔	┘	Θ	∙
-A	◙	→	*	:	J	Z	j	z		è	Ü	¬	║	╩	┌	Ω	·
-B	♂	←	+	;	K	[	k	{		ï	¢	½	╗	╦	■	δ	√
-C	♀	∟	,	<	L	\	l	|		î	£	¼	╝	╠	▄	∞	n
-D	♪	↔	-	=	M	]	m	}		ì	¥	¡	╜	═	▌	ϕ	2
-E	♫	▲	.	>	N	^	n	~		Ä	Pt	«	╛	╬	▐	ε	■
-F	☼	▼	/	?	O	_	o	⌂		Å	ƒ	»	┐	╧	▀	∩	

Code Page 850 (Multilingual)

Hex Digits 1st → 2nd ↓	0-	1-	2-	3-	4-	5-	6-	7-		8-	9-	A-	B-	C-	D-	E-	F-
-0		►		0	@	P	`	p		Ç	É	á	▒	└	ð	Ó	-
-1	☺	◄	!	1	A	Q	a	q		ü	æ	í	▓	┴	Ð	ß	±
-2	☻	↕	"	2	B	R	b	r		é	Æ	ó	▓	┬	Ê	Ô	=
-3	♥	!!	#	3	C	S	c	s		â	ô	ú	│	├	Ë	Ò	¾
-4	♦	¶	$	4	D	T	d	t		ä	ö	ñ	┤	─	È	õ	¶
-5	♣	§	%	5	E	U	e	u		à	ò	Ñ	Á	┼	ı	Õ	§
-6	♠	▬	&	6	F	V	f	v		å	û	ª	Â	ã	Í	µ	÷
-7	•	↨	'	7	G	W	g	w		ç	ù	º	À	Ã	Î	þ	¸
-8	◘	↑	(	8	H	X	h	x		ê	ÿ	¿	©	╚	Ï	Þ	°
-9	○	↓	)	9	I	Y	i	y		ë	Ö	®	╣	╔	┘	Ú	¨
-A	◙	→	*	:	J	Z	j	z		è	Ü	¬	║	╩	┌	Û	•
-B	♂	←	+	;	K	[	k	{		ï	ø	½	╗	╦	▀	Ù	¹
-C	♀	∟	,	<	L	\	l	|		î	£	¼	╝	╠	▄	ý	³
-D	♪	↔	-	=	M	]	m	}		ì	Ø	¡	¢	═	¦	Ý	²
-E	♫	▲	.	>	N	^	n	~		Ä	×	«	¥	╬	Ì	¯	■
-F	☼	▼	/	?	O	_	o	⌂		Å	ƒ	»	┐	¤	▀	´	

Code Page 860 (Portuguese)

Hex Digits 1st → 2nd ↓	0-	1-	2-	3-	4-	5-	6-	7-		8-	9-	A-	B-	C-	D-	E-	F-
-0		►		0	@	P	`	p		Ç	É	á	▒	└	╨	α	≡
-1	☺	◄	!	1	A	Q	a	q		ü	À	í	▒	┴	╤	β	±
-2	☻	↕	"	2	B	R	b	r		é	È	ó	▒	┬	╥	Γ	≥
-3	♥	!!	#	3	C	S	c	s		â	ô	ú	│	├	╙	π	≤
-4	♦	¶	$	4	D	T	d	t		ã	õ	ñ	┤	─	╘	Σ	⌠
-5	♣	§	%	5	E	U	e	u		à	ò	Ñ	╡	┼	╒	σ	⌡
-6	♠	▬	&	6	F	V	f	v		Á	Ú	ª	╢	╞	╓	µ	÷
-7	•	↨	'	7	G	W	g	w		ç	ù	º	╖	╟	╫	τ	≈
-8	◘	↑	(	8	H	X	h	x		ê	Ì	¿	╕	╚	╪	Φ	°
-9	○	↓	)	9	I	Y	i	y		Ê	Õ	Ò	╣	╔	┘	Θ	•
-A	◙	→	*	:	J	Z	j	z		è	Ü	¬	║	╩	┌	Ω	·
-B	♂	←	+	;	K	[	k	{		Ì	¢	½	╗	╦	■	δ	√
-C	♀	∟	,	<	L	\	l	|		Ô	£	¼	╝	╠	▄	∞	n
-D	♪	↔	-	=	M	]	m	}		ì	Ù	¡	╜	═	▌	ϕ	2
-E	♫	▲	.	>	N	^	n	~		Ã	Pt	«	╛	╬	▐	ε	■
-F	☼	▼	/	?	O	_	o	⌂		Â	Ó	»	┐	╧	▀	∩	

Code Page 863 (Canadian-French)

Hex Digits 1st → 2nd ↓	0-	1-	2-	3-	4-	5-	6-	7-		8-	9-	A-	B-	C-	D-	E-	F-
-0		►		0	@	P	`	p		Ç	É	¦	▒	└	╨	α	≡
-1	☺	◄	!	1	A	Q	a	q		ü	È	´	▓	┴	╤	β	±
-2	☻	↕	"	2	B	R	b	r		é	Ê	ó	▓	┬	╥	Γ	≥
-3	♥	!!	#	3	C	S	c	s		â	ô	ú	│	├	╙	π	≤
-4	♦	¶	$	4	D	T	d	t		Â	Ë	¨	┤	─	╘	Σ	⌠
-5	♣	§	%	5	E	U	e	u		à	Ï	˜	╡	┼	╒	σ	⌡
-6	♠	▬	&	6	F	V	f	v		¶	û	³	╢	╞	╓	µ	÷
-7	•	↨	'	7	G	W	g	w		ç	ú	¯	╖	╟	╫	τ	≈
-8	◘	↑	(	8	H	X	h	x		ê	¤	Î	╕	╚	╪	Φ	°
-9	○	↓	)	9	I	Y	i	y		ë	Ô	⌐	╣	╔	┘	Θ	•
-A	◙	→	*	:	J	Z	j	z		è	Ù	¬	║	╩	┌	Ω	·
-B	♂	←	+	;	K	[	k	{		ï	¢	½	╗	╦	█	δ	√
-C	♀	∟	,	<	L	\	l	|		î	£	¼	╝	╠	▄	∞	n
-D	♪	↔	-	=	M	]	m	}		=	Ù	¾	╜	═	▌	ϕ	2
-E	♫	▲	.	>	N	^	n	~		À	Û	«	╛	╬	▐	ε	■
-F	☼	▼	/	?	O	_	o	⌂		§	ƒ	»	┐	╧	▀	∩	

Code Page 865 (Nordic)

Hex Digits 1st → 2nd ↓	0-	1-	2-	3-	4-	5-	6-	7-		8-	9-	A-	B-	C-	D-	E-	F-
-0		►		0	@	P	`	p		Ç	É	á	▒	└	╨	α	≡
-1	☺	◄	!	1	A	Q	a	q		ü	æ	í	▒	┴	╤	β	±
-2	☻	↕	"	2	B	R	b	r		é	Æ	ó	▒	┬	╥	Γ	≥
-3	♥	!!	#	3	C	S	c	s		â	ô	ú	│	├	╙	π	≤
-4	♦	¶	$	4	D	T	d	t		ä	ö	ñ	┤	─	╘	Σ	⌠
-5	♣	§	%	5	E	U	e	u		à	ò	Ñ	╡	┼	╒	σ	⌡
-6	♠	▬	&	6	F	V	f	v		å	û	ª	╢	╞	╓	µ	÷
-7	•	↨	'	7	G	W	g	w		ç	ú	º	╖	╟	╫	τ	≈
-8	◘	↑	(	8	H	X	h	x		ê	ÿ	¿	╕	╚	╪	Φ	°
-9	○	↓	)	9	I	Y	i	y		ë	Ö	⌐	╣	╔	┘	Θ	•
-A	◙	→	*	:	J	Z	j	z		è	Ü	¬	║	╩	┌	Ω	·
-B	♂	←	+	;	K	[	k	{		ï	ø	½	╗	╦	■	δ	√
-C	♀	∟	,	<	L	\	l	|		î	£	¼	╝	╠	▄	∞	n
-D	♪	↔	-	=	M	]	m	}		ì	Ø	¡	╜	═	▌	ϕ	2
-E	♫	▲	.	>	N	^	n	~		Ä	Pt	«	╛	╬	▐	ε	■
-F	☼	▼	/	?	O	_	o	⌂		Å	ƒ	¤	┐	╧	▀	∩	

Appendix G. Messages

This appendix contains DOS mode messages that do not have online help.

Messages are listed in **bold** type and the explanation and action follow the message. The explanation tells you whether a command, program, or the system generated the message. In some cases, the message is generated by several different programs or commands. In these cases, the first word after the explanation heading is COMMANDS. Where the message is generated by an internal DOS mode file, the first word is SYS.

DOS Mode Messages

Access denied

Explanation: COMMANDS

Executing the requested command violates the access mode of the file, subdirectory, or device involved. For example, an attempt was made to write to a file marked read-only, or read a file that is write-only. Or an attempt was made to open a subdirectory as a file.

Action:

Use a different filename or, if the file is read-only and you need to use that file, change its attribute with the ATTRIB command.

Bad command or filename

Explanation: SYS

The command you entered is not a valid DOS mode command.

Action:

- Check the spelling of the command and re-enter it.
- If the command name is spelled correctly, check to see that the default drive contains the external command or batch file you are trying to execute.

Batch file missing

Explanation: SYS

DOS could not locate the batch file it was processing. The file may have been erased or renamed by one of the steps within it. The current drive may have been changed within the batch file, the batch processor can no longer find the .BAT file using your PATH. Batch processing stops and the DOS mode prompt appears.

Action:

- If the filename was changed, correct the command that changed the name.
- If the file was erased, use your backup copy. If you used EDLIN to create the file or make changes, rename the .BAK file to .BAT. Correct the command that deleted the file.
- Include the drive letters in the PATH.

BREAK is on|off

Explanation: BREAK

This message indicates the status of BREAK, either on or off.

Action:

Enter the command you want. For example, if the screen shows **Break is off** and **Break is on** is desired, enter the command:

```
BREAK ON
```

Code page *xxx* not prepared for all devices

Explanation: CHCP

CHCP was unable to select the code page for one of the following reasons:

- A device was not prepared for the requested code page.
- A device I/O error occurred.
- The device is currently printing.
- The device does not support code page switching.

Action:

Ensure there is a DEVINFO statement in the CONFIG.SYS file for each device, for example, printer, keyboard, and screen.

Code page *xxx* not prepared for system

Explanation: CHCP

CHCP was unable to select the specified code page.

Action:

Verify that the CODEPAGE in CONFIG.SYS includes the code page you are attempting to prepare.

Cannot do binary reads from a device

Explanation: COPY

You used the **/B** parameter with a device name while trying to copy from the device. The copy cannot be performed in binary mode because COPY must be able to detect the end-of-file from the device.

Action:

Re-enter COPY and omit the **/B** parameter or use the **/A** parameter after the device name.

Cannot load COMMAND, this session halted

Explanation: SYS

The system attempted to reload the command processor, but the area in which the system keeps track of available storage was destroyed; or the command processor was not found in the path specified by the COMSPEC environment variable. OS/2 sessions may still be active.

Action:

If DOS mode is required, terminate all OS/2 sessions and ensure that COMMAND.COM is present on the diskette or drive you are using and restart the system.

Cannot start COMMAND, exiting

Explanation: SYS

While the system was attempting to load another copy of the command processor, it did not find enough available storage to contain the new copy of COMMAND.COM. OS/2 sessions may still be active.

Action:

- Terminate all OS/2 sessions that may be running.
- Increase the RMSIZE value in your CONFIG.SYS file. The changed RMSIZE in active only after restarting the system.
- Restart the system.

Content of destination lost before copy

Explanation: COPY

An illegal concatenation of files was detected during the copy command. The destination filename was included among the source files being concatenated. The condition was not detectable until the destination file had been copied over.

Action:

- Restore damaged files from your backup copy.
- Information message, no correction possible. Review the syntax of the COPY command to prevent reoccurence.

CTTY not supported In OS/2

Explanation: SYS

The DOS command CTTY was attempted.

Action:

None, informational message.

Current drive is no long valid

Explanation: COMMANDS

While attempting to get the current drive for the DOS mode prompt, COMMAND found that the drive is no longer valid.

Action:

Change your current drive to a valid drive.

Duplicate filename or file not found

Explanation: RENAME

You tried to rename a file to a filename that already exists on the disk, or the file to be renamed could not be found on the specified (or default) drive. RENAME is warning you that you are using the same name for two files, or else it cannot find the file you are trying to rename.

Action:

Check to see if you typed the filename correctly. Re-enter the RENAME command.

ECHO is on|off

Explanation: ECHO

This message indicates the status of ECHO, either on or off.

Action:

Enter the command you want. For example if the screen shows **Echo is off** and **Echo is on** is desired, enter the command:

```
ECHO ON
```

Error in EXE file

Explanation: SYS

An error was detected in the relocation information placed in the file by the LINK program. This may be due to a modification to the file.

Action:

- If you are using a purchased program, rerun the program using your backup copy.
- If you are using a program you wrote yourself, go through the LINK procedure again.

Error writing to device

Explanation: COMMANDS

Informational message. The system encountered an I/O error when writing output to a device. The device is unable to handle the number of bytes requested.

Action:

Change the amount of data in the file and retry the command.

EXEC failure

Explanation: COMMANDS

The system encountered an error while reading a command or other program from disk.

Action:

Try the command again, if the error reappears, attempt to load from a backup copy.

File allocation table bad, drive *x*
Abort, Retry, Ignore?

Explanation: SYS

The file allocation has been corrupted.

Action:

- Retry.
- Restart the system.
- Reformat drive *x* and restore any lost files from your backup copy.

File cannot be copied onto itself

Explanation: COPY or XCOPY

You tried to COPY a file and place the copy (with the same name as the original) in the same directory and on the same disk as the original file.

Action:

Change the name given to the copy, or put it in a different directory, or put it on another disk.

File creation error

Explanation: SYS and COMMANDS

An unsuccessful attempt was made to add a new filename to the directory or to replace a file that was already there.

Action:

If the file was already there, check to see if the file is marked "read-only" and cannot be replaced. Otherwise, run CHKDSK to determine if the directory is full, or if some other condition caused the error.

File not found

Explanation: SYS and COMMANDS

A file named in a command or command parameter does not exist in the directory of the specified (or default) drive.

Action:

Retry the command using the correct filename.

FOR cannot be nested

Explanation: Batch file

More than one FOR subcommand was found on one command line in the batch file.

Action:

Use only one FOR subcommand per command line. Then retry the command.

Insert disk with batch file
and press any key when ready

Explanation: SYS

The diskette that contained the batch file being processed was removed. The batch processor is trying to find the next command in the file.

Action:

Insert the diskette in the appropriate drive and press any key. Processing will continue.

Insert disk with \COMMAND.COM in drive *x*
and press any key when ready

Explanation: SYS

The system is attempting to reload the command processor, but COMMAND.COM is not in the drive indicated by the COMSPEC environment variable.

Action:

Insert the diskette that has COMMAND.COM in the indicated drive and press any key.

Insert diskette for drive *x* and press any key when ready

Explanation: SYS.

In a system with logical drives, a drive that is not the current default drive is being referenced, so DOS is asking for the diskette corresponding to that drive.

Action:

If the diskette for *x* is different than the one currently in the drive, insert the appropriate diskette and press any key.

Insufficient disk space

Explanation: SYS and COMMANDS

The disk does not contain enough free space to contain the file being written.

Action:

If you suspect this condition is invalid, run CHKDSK to determine the status of the disk. Otherwise, use another disk and retry the command.

Insufficient memory

Explanation: COMMANDS

The amount of available storage is too small to allow these commands to function.

Action:

Increase the value for the RMSIZE statement and/or decrease the value for the BUFFERS statement in the CONFIG.SYS file. Restart the system and try the command again. If the message still appears, your system does not have enough storage to execute the command.

Intermediate file error during pipe

Explanation: SYS

The system is unable to create one or both of its intermediate files because the default drive's root directory was full, or the system is unable to locate the piping files, or the disk does not have enough space to hold the data being piped.

Action:

Erase some files from the default drive's root directory, and reissue the command that failed. If you get the same message, one of the programs in the command line has erased one or both of the piping files. Correct the program and reissue the command line.

Invalid COMMAND.COM in drive *n*

Explanation: SYS

When the system tried to reload the command processor, the copy of COMMAND.COM on the disk was found to be an incorrect version.

Action:

Restore the correct COMMAND.COM to the indicated drive.

Invalid date

Explanation: SYS

You entered an invalid date or delimiter. The only valid delimiters in a date entry are hyphens (-), slashes (/), and periods (.), or the date separator defined for the country specified by the COUNTRY statement in CONFIG.SYS.

Action:

Re-enter a valid date.

Invalid directory

Explanation: SYS and COMMANDS

One of the directories in the specified path does not exist.

Action:

Retry the command using a valid directory.

Invalid disk change

Explanation: SYS

The diskette in the drive was changed while files were still open on the diskette.

Action:

Reinsert the correct diskette.

Invalid drive in search path

Explanation: SYS

An invalid drive specifier was found in one of the paths specified in the PATH command. This message appears when DOS attempts to locate a command or batch file rather than at the time the incorrect PATH command was issued.

Action:

1. Enter PATH. This displays the paths previously defined.
2. Find the invalid specifier.
3. Re-enter the PATH command with the valid drive specifier and the desired paths.

Invalid drive specification

Explanation: COMMANDS

An invalid or non-existent drive specification was just entered in the command or in one of its parameters, or the source and target drive are the same.

Action:

Re-enter the command using a valid drive specifier.

Invalid filename or file not found

Explanation: RENAME or TYPE

You tried to rename a file that was either invalid or not found in the specified directory. TYPE does not allow global filename characters.

Action:

Enter the correct filename.

Invalid number of parameters

Explanation: COMMANDS

You have specified too few or too many parameters for the command you issued.

Action:

Check the syntax of the command and re-enter the command.

Invalid parameter

Explanation: SYS and COMMANDS

One or more of the parameters entered for these commands are not valid or have been placed in the wrong order.

Action:

If the program expects a drive specifier, enter a colon (:) following the drive letter. In other cases, make sure the character following the slash (/) is valid for the program being run.

Invalid path, not directory
or directory not empty

Explanation: RMDIR

- The specified directory was not removed because one of the names you specified in the path was not a valid directory name.
- The directory you specified still contains entries for files or other subdirectories (with the exception of the . and .. entries). It is possible that there are hidden files in the directory. The CHKDSK command will detect them.
- You cannot remove a current directory.

Action:

Try one of the following:

- Correct the invalid directory name in the path.
- Delete any files or remove any subdirectories in the directory.
- Change to a different subdirectory and try again.
- Run CHKDSK *.* in the directory. If there are hidden files, erase them according to the instructions of the application that created them.

Invalid path or filename

Explanation: ATTRIB or COPY

You specified a directory or filename that does not exist.

Action:

Use the correct name. Retry the command after checking for the following:

- Correct spelling of names
- Valid directory names
- Existence of file in the subdirectory specified.

Invalid time

Explanation: TIME

An invalid time or delimiter was entered.

Action:

Re-enter the correct time. The only valid delimiters are:

- Colon (:) between the hours and minutes
- Colon (:) between the minutes and seconds
- Period (.) between the seconds and hundredths of a second.

Label not found

Explanation: Batch file

Informational message. A GOTO command named a label that does not exist in the batch file. This caused the system to read to the end of the batch file, ending batch processing.

Action:

If you do not want the GOTO to exit the batch file, edit the batch file and put the label in the desired location.

Lock violation

Explanation: XCOPY

A source file has part or all of it locked against reading.

Action:

Wait a short time and try again.

Memory allocation error
Cannot load COMMAND, this session halted

Explanation: SYS

The system attempted to reload the command processor, but the area in which the system keeps track of available storage was destroyed; or the command processor was not found in the path specified by the COMSPEC environment variable. OS/2 sessions may still be active.

Action:

If DOS mode is required, terminate all OS/2 sessions and ensure that COMMAND.COM is present on the diskette or drive you are using and restart the system.

Must specify ON or OFF

Explanation: BREAK or VERIFY

You entered something other than on or off.

Action:

Try again, specifying on or off.

No free file handles
Cannot start COMMAND, exiting

Explanation: SYS

An attempt to load a second copy of the command processor failed because there are too many file handles opened throughout the system.

Action:

Terminate applications that are using OS/2 sessions to reduce the number of open files.

No path

Explanation: PATH

Informational message. An alternate path for DOS to search for commands and batch files is not specified.

Action:

Informational message unless you want to define a set of paths. If so, enter PATH and the set of paths you want. Then press Enter.

Out of environment space

Explanation: SYS

Informational message. DOS was unable to accept the SET command you just issued because it was unable to expand the area in which the environment information is kept. This normally occurs when you try to add to the environment after loading a program which makes itself resident (PRINT or MODE for example).

Action:

Edit CONFIG.SYS to include or increase the /E: parameter in the SHELL statement. Then restart the system.

Path not found

Explanation: SYS and COMMANDS

A file or path named in a command or command parameter does not exist in the directory of the specified (or default) drive.

Action:

Retry the command using the correct path and filename.

Program too big to fit in memory

Explanation: SYS

The file containing the external command cannot be loaded because it is larger than the available free storage.

Action:

Increase the number in the RMSIZE statement in your CONFIG.SYS file to a larger value and/or decrease the value for the BUFFERS statement in the CONFIG.SYS file. Restart your system and reissue the command. If the message reappears, your system does not have enough storage to execute the command.

Syntax error

Explanation: SYS

The command format you typed is incorrect.

Action:

Make sure that you have used the correct format for this command.

Terminate batch job (Y/N)?

Explanation: SYS

This message appears when you press Ctrl-Break while DOS is processing a batch file.

Action:

Press Y to stop processing the batch file. Pressing N only ends the command that was executing when Ctrl-Break was pressed; processing resumes with the next command in the batch file.

Top level process aborted, cannot continue

Explanation: SYS

The system attempted to reload the command processor, but the area in which the system keeps track of available storage was destroyed; or the command processor was not found in the path specified by the COMSPEC environment variable. OS/2 sessions may still be active.

Action:

If DOS mode is required, terminate all OS/2 sessions and ensure that COMMAND.COM is present on the diskette or drive you are using and restart the system.

Unable to create directory

Explanation: SYS and COMMANDS

- The directory you want to create already exists.
- One of the directory path names you specified could not be found.
- You attempted to add a directory to the root directory and it is full.
- A file by that name already exists in that directory.
- The directory name you specified contains invalid characters or is a reserved device name.

Action:

Do the following:

- Check to see if a directory by that name exists in the parent directory (or current directory).
- Recheck all your directory names to make sure they are valid.
- Use CHKDSK to see if your directory is full.

VERIFY is on|off

Explanation: VERIFY

This message indicates the status of VERIFY, either on or off.

Action:

Enter the command you want. For example if the screen shows **Verify is off** and **Verify is on** is desired, enter the command:

```
VERIFY ON
```

Appendix H. OS/2 Diskettes and Directories

This appendix describes the contents of the IBM Operating System/2™ (OS/2™) diskettes and the organization of the directories created on your fixed disk by the system installation program.

Operating System/2 Diskettes

There are four Operating System/2 diskettes:

- *Diskette 1* contains the files necessary to start the operating system and the Program Selector. The system can be started by inserting this diskette in Drive A and then pressing Ctrl+Alt+Del.
- *Diskette 2* and *Diskette 3* contain OS/2 external commands, device drivers, and related support files.
- The *Installation Diskette* contains the installation program that installs OS/2. The installation program is started by inserting this diskette in drive A and then pressing Ctrl+Alt+Del.

Directories Created by the Installation Program

Once you have installed OS/2 on your fixed disk with the system installation program, your fixed disk has the directory structure described on the pages that follow.

The Root Directory

Your root directory (\) contains the system files necessary to start Operating System/2 and the Program Selector.

The system files in your root directory are:

ABIOS.SYS
ANSICALL.DLL
AUTOEXEC.BAT
BKSCALLS.DLL
BMSCALLS.DLL
BVSCALLS.DLL
CLOCK01.SYS
CLOCK02.SYS
CMD.EXE
COMMAND.COM
CONFIG.SYS
COUNTRY.SYS
CPISPFPC.DLL
DISK01.SYS
DISK02.SYS
DMPC.EXE
DOSCALL1.DLL
DTM.DLL
F80000.BIO
F80100.BIO
FC0400.BIO
FC0500.BIO
FORMATS.TBL
HARDERR.EXE
INSTALL.LOG
ISPD.MSG
ISPM.MSG
KBD01.SYS
KBD02.SYS
KBDCALLS.DLL
KEYBOARD.DCP
MONCALLS.DLL
MOUCALLS.DLL
MSG.DLL
NLS.DLL
OSO001.MSG
OSO001H.MSG
OS2INIT.CMD
PRINT01.SYS
PRINT02.SYS
QUECALLS.DLL
4201.DCP
5202.DCP
SCREEN01.SYS
SCREEN02.SYS
SESMGR.DLL
SHELL11F.AIF
SHELL11F.AII
SHELL11F.CNF
SHELL11F.EXE
SHELL11F.LIB
SHELL11F.PRO
SPOOLCP.DLL
STXTDMPC.DLL
SWAPPER.DAT
SWAPPER.EXE
SYSBAS.EXE
SYSDISP.CMD
VIOCALLS.DLL
VIOTBL.DCP

Of these system files located in your root directory, the following are system dependent base device drivers:

Family 1 **PC AT, PC XT 286**	**Family 2** **PS/2**
CLOCK01.SYS	CLOCK02.SYS
DISK01.SYS	DISK02.SYS
KBD01.SYS	KBD02.SYS
PRINT01.SYS	PRINT02.SYS
SCREEN01.SYS	SCREEN02.SYS

Only the base device driver files listed under the family of your computer are necessary to start your system. You do not need the base device driver files listed under the other family.

Warning: These device drivers cannot be specified by you with DEVICE statements in CONFIG.SYS.

The following files are used by the system installation program and are not necessary to start the system:

INSTALL.LOG
SYSBAS.EXE
SYSDISP.CMD

The following files are Family 2 specific and are not necessary if your computer is in Family 1:

ABIOS.SYS
F80000.BIO
F80100.BIO
FC0400.BIO
FC0500.BIO

The operating system files in your root directory that are necessary for the proper starting and executing of OS/2 should not be erased. Should you inadvertently erase a needed system file, you can copy it to your root directory using this procedure:

1. Insert Operating System/2 Diskette 1 into drive A and press Ctrl+Alt+Del. The system is started and the Program Selector menu is displayed.
2. Select either command prompt from the menu.
3. If necessary, remove Diskette 1 and insert the diskette that contains the system file to be copied.
4. From the command prompt enter:

```
COPY sysfile.ext C:\
```

Other Directories Created by System Installation

The \OS2 directory contains OS/2 external commands and installable device drivers. Subdirectories to this directory are:

- The \OS2\INTRO directory, which contains an online introduction to Operating System/2.
- The \OS2\INSTALL directory, which contains application installation utilities.

The \SPOOL directory is where temporary spool files needed by the print spooler are stored.

Glossary

A

absolute (priority). Pertaining to a priority (of a process) that is not varied by the operating system. Contrast with dynamic (priority).

active program. A program that is running. An active program can be interactive (running and receiving input from the user) or noninteractive (running but not receiving input from the user). Contrast with inactive program.

alias. An alternate label or name.

alternate code page. The code page that replaces the primary code page when the user switches code pages.

C

child process. (1) A process that is created by another process. (2) A dependent process. Contrast with parent process.

COMMAND.COM. The program that interprets and runs DOS commands.

CMD.EXE. The program that interprets and runs OS/2 commands.

code page. A defined set of alphanumeric, graphic, and control characters.

command processor. In OS/2, a program (COMMAND.COM or CMD.EXE) that interprets and runs commands.

command prompt. A displayed symbol, such as C>, that requests input from the user.

coprocessor. A microprocessor that extends the address range of the processor in the system unit or adds specialized instructions to handle a particular category of operations; for example, an I/O coprocessor, a math coprocessor, a networking coprocessor.

D

dump diskette. A diskette (created using the CREATEDD command) that contains the contents of storage at a specified point in time.

dynamic linking. The delayed linking of a program to a routine so that the routine is not linked until load time or run time.

dynamic (priority). Pertaining to a priority (of a process) that is varied by the operating system. Contrast with absolute (priority).

dynamic link library. A module that is linked at load time or run time.

E

extended partition. That area beyond the primary partition. This area can be divided into smaller areas or partitions, each of which can be assigned a drive letter and be accessed as though it were a separate fixed disk.

extended storage. Storage that supplements system storage.

F

FCB (file control block). A record that contains all of the information about a file; for example, its structure, length, and name.

I

inactive program. A program that has been started but is suspended so that it is no longer running. Contrast with active program.

interactive session. That group of processes (or tasks) that currently own the keyboard, mouse, and display. or other interactive input device.

interprocess communication. The exchange of information (such as, by means of messages) between processes or threads.

L

local variable. A parameter that is defined and used only within a specified portion of the program in which it is declared.

logical device. An input/output device identified in a program by a label or number that corresponds to the actual label or number assigned to the device. Contrast with physical device.

M

major event code. A code assigned to major system events, such as opening a file, writing to a file, and sending output to the display. Major event code values range from 0 to 255.

multilingual code page. The code page that contains the alphanumeric and special symbols used in most European, North American, and South American countries. In OS/2, this is code page 850.

multiprogramming. The concurrent processing of two or more programs by a computer.

multitasking. The virtually simultaneous processing of two or more tasks by the operating system.

N

national language code page. The code page that contains the alphanumeric and special symbols used in a particular language.

nonescaping key. A key that allows a character to be typed without the print position being changed.

noninteractive program. An active program that is running but cannot receive input from the keyboard or other interactive input device. Contrast with interactive program.

normal priority process. A process for which a designated time slice is sufficient for the process to be completed.

O

operating system files. Files (such as, IBMBIO.COM and IBMDOS.COM) that contain the operating system.

P

parent process. (1) A process that creates other processes. (2) An independent process. Contrast with child process.

partition. An area of storage on a fixed disk that contains a particular operating system or logical drives where data and programs can be stored.

physical device. An input/output device identified in a program by its actual label or number. Contrast with logical device.

primary code page. The code page that is active when the computer is started.

primary partition. The partition that contains the operating system that is active when the computer is started.

print spooler. A program that allows output to be temporarily stored until it can be printed.

process. A collection of system resources including one or more threads.

process identification number. A number assigned by the operating system to each program that is started. The number is used internally by programs when they need to communicate with each other.

processor. In a computer, a functional unit that interprets and processes instructions.

R

RAS programs. Reliability, Availability, and Serviceability programs. These programs monitor the operating system and facilitate problem determination.

refresh (the screen). To update (the screen) with the latest information.

replaceable parameter. A parameter whose value is supplied when a batch file is run or when it is entered at the OS/2 command prompt.

round robin scheduling. The process of executing programs in a sequence, with each program being allocated a specific amount of time.

run time. The phase of program execution during which program instructions are being performed.

S

segment. A variable-length area of contiguous storage addresses not exceeding 64K bytes.

session. In this book, this term refers to that group of processes (or tasks) associated with an application.

spooling. Reading and writing input and output on an intermediate device in a format convenient for later processing or output.

starvation. A situation in which a process cannot complete its designated task because it is kept waiting too long for processing time.

storage allocation. The use of operating system functions to assign storage areas to tasks.

storage compaction. Relocating allocated storage segments into contiguous locations in order to place all free storage in one large block.

storage dump. A copy of the contents of a storage or a part of a storage at a specified point in time, that is recorded on an external medium such as a disk.

suspend. An action that causes an active program to become temporarily inactive. In effect, the suspended program is waiting for the user to reactivate it.

swap file. A file that contains segments of a program or data temporarily moved out of main storage.

swapping. A process that moves segments between system storage and extended storage.

system extensions. Additional functions added to the operating system, such as data base management or communications function.

system storage. Program-addressable storage from which instructions and other data can be loaded directly into registers for subsequent processing. Contrast with extended storage.

system trace. A historical record of specific events in the execution of the operating system. The record is usually produced for debugging purposes.

T

task. (1) A basic unit of work to be accomplished by the computer. (2) A program running in the computer.

thread. The smallest unit of processing within a process.

time-critical process. A high-priority process that must be run ahead of other processes in a queue.

time slice. A designated interval of time allocated for processing a task.

Index

Special Characters

. (current directory) 2-115
.. (parent directory) 2-26, 2-115
< (redirect input) 5-2
() (command grouper) 5-4
+ (combine files) 2-41
| (piping symbol) 5-2
|| (OR command operator) 5-4
& (command separator) 5-4
&& (AND command operator) 5-3
$$$ EDLIN filename extension C-1
% and %% (for replaceable parameters) 3-9
> (redirect output) 5-1
>> (append redirected output) 5-2
¦ (piping symbol) 5-2
¦¦ (OR command operator) 5-4

A

abort read/write operation G-2
accent characters 2-82, E-1
active partitions 2-64
allocation of storage 1-2
AND command operator (&&) 5-3
ANSI command 2-7
ANSI.SYS device driver 4-16
APPEND command 2-8
appending files 2-39
applying patches 2-98
archive bit of a file 2-14
ASCII (text) file copying 2-39
ASCII character 124 5-2
ASCII comparison 2-36
ASSIGN command 2-12
asynchronous communications device driver 4-17
asynchronous communications mode 2-89
ATTRIB command 2-14
Australia
  code pages supported 4-11
  country code 4-11
AUTOEXEC.BAT file 3-5
automatic patching 2-99
automatic program execution 3-5
automatic starting of OS/2 programs 3-6

B

backing up files 2-17
backing up subdirectories 2-17
backing up the disk 2-17
BACKUP command 2-17
backup files
  creating 2-17
  restoring 2-112
backup log file 2-17
BACKUP.LOG file 2-17
BAK EDLIN filename extension C-2, C-15, C-24
BAT extension for DOS batch file 3-1
batch commands 3-10
  CALL 3-11
  ECHO 3-13
  ENDLOCAL 3-15
  EXTPROC 3-17
  FOR 3-18
  GOTO 3-20
  IF 3-21
  PAUSE 3-24
  REM (remarks) 3-26
  SETLOCAL 3-27
  SHIFT 3-28
batch file extension
  DOS mode 3-1
  OS/2 mode 3-1

batch filenames 3-1
batch files
  calling other batch files 3-3
  chaining 3-3
  displaying remarks in 3-26
  ending processing of 2-21, 3-24
  interrupting 3-3
  nesting 3-3
  running other batch files 3-3
  suppressing echoing of state-
    ments in 3-10
  suspending processing of 3-24
batch processing 3-1
  DOS mode 3-1
  OS/2 mode 3-1
Belgium
  code pages supported 4-11
  country code 4-11
  keyboard 2-81
binary comparison 2-36
binary file copying 2-40
block device drivers B-8
bootable system diskette H-1
BREAK
  CONFIG.SYS statement 4-5
  DOS command 2-20
BUFFERS statement in
  CONFIG.SYS 4-7

C

cache support 4-34
CALL batch command 3-11
calling batch files 3-3
Canada (English)
  code pages supported 4-11
  country code 4-11
Canada (French)
  code pages supported 4-11
  country code 4-11
  keyboard 2-81
Canadian-French code page F-5
CD command 2-26
chaining batch files 3-3
changing a search path
  APPEND 2-8
  DPATH 2-59
  PATH 2-101
changing code pages 2-22
changing directories 2-26
changing file attributes 2-14
changing filenames 2-109
changing keyboards 2-81
changing statements in
  CONFIG.SYS 4-2
changing the clock
  DATE command 2-45
  TIME command 2-137
changing the current
  directory 2-26
changing the disk name 2-84
changing the prompt 2-105
characters per line 2-94
CHCP command 2-22
CHDIR command 2-26
CHKDSK command 2-28
clearing the screen 2-31
clock, resetting 2-45, 2-137
CLS command 2-31
CMD command 2-32
CMD extension for OS/2 batch
  file 3-1
code page switching 2-22
  interrelated statements in
    CONFIG.SYS 4-3
  preparing devices for 4-30
  purpose of 2-23
code pages 4-8
  definition 2-23
  names of 4-8
  preparing devices to use 4-30
  specifying with CODEPAGE 4-8
  supported by OS/2 4-8
  437 (U.S. IBM PC) F-2
  850 (multilingual) F-3
  860 (Portugal) F-4
  863 (Canadian-French) F-5

code pages *(continued)*
865 (Nordic) F-6
CODEPAGE statement in CONFIG.SYS 4-8
collating sequence used by SORT 4-10
COM port in DOS mode 2-121
combining files 2-39
combining segments 1-3
COMMAND command 2-34
command diagram 2-3
external commands 2-3
internal commands 2-3
command diagrams, reading 2-2
command length 2-2
command operators 5-1
command parameters 2-2
command processor
DOS mode 1-1, 2-34
OS/2 mode 1-1, 2-32
command prompt
changing the default 2-105
for DOS mode 2-105
for OS/2 mode 2-105
command separator symbol 5-4
command symbols
DOS mode 1-1
OS/2 mode 1-1
commands 2-2, 3-10
ANSI 2-7
APPEND 2-8
ASSIGN 2-12
ATTRIB 2-14
BACKUP 2-17
batch 3-10
CALL 3-11
ECHO 3-13
ENDLOCAL 3-15
EXTPROC 3-17
FOR 3-18
GOTO 3-20
IF 3-21
PAUSE 3-24
REM (remarks) 3-26
SETLOCAL 3-27
SHIFT 3-28
commands *(continued)*
BREAK 2-20
CHCP 2-22
CHDIR 2-26
CHKDSK 2-28
CLS 2-31
CMD 2-32
COMMAND 2-34
COMP 2-36
CONFIG.SYS
BREAK 4-5
BUFFERS 4-7
CODEPAGE 4-8
COUNTRY 4-10
DEVICE 4-13
DEVINFO 4-30
DISKCACHE 4-34
FCBS 4-36
IOPL 4-38
LIBPATH 4-39
MAXWAIT 4-41
MEMMAN 4-42
PAUSEONERROR 4-44
PRIORITY 4-45
PROTECTONLY 4-46
PROTSHELL 4-47
REM (remarks) 4-49
RMSIZE 4-50
RUN 4-51
SHELL 4-53
SWAPPATH 4-55
THREADS 4-56
TIMESLICE 4-57
TRACE 4-59
TRACEBUF 4-61
COPY 2-39
CREATEDD 2-43
DATE 2-45
DEL (see ERASE)
DETACH 2-47
DIR 2-49
DISKCOMP 2-53
DISKCOPY 2-56
DPATH 2-59
ERASE 2-61

commands *(continued)*
EXIT 2-63
FDISK 2-64
FIND 2-67
FORMAT 2-70
GRAFTABL 2-74
HELP 2-76
how to enter 2-2
JOIN 2-79
KEYB 2-81
LABEL 2-84
MKDIR 2-86
MODE 2-88
MORE 2-97
PATCH 2-98
PATH 2-101
PRINT 2-103
PROMPT 2-105
RECOVER 2-107
REN 2-109
REPLACE 2-110
RESTORE 2-112
RMDIR 2-115
SET 2-117
SETCOM40 2-121
SORT 2-124
SPOOL 2-126
START 2-130
SUBST 2-133
SYS 2-135
TIME 2-137
TRACE 2-139
TRACEFMT 2-141
TREE 2-143
TYPE 2-144
VER 2-146
VERIFY 2-147
VOL 2-149
XCOPY 2-151
comments
displaying by a batch file 3-24
in a batch file 3-26
in CONFIG.SYS file 4-49
COMP command 2-36
comparing diskettes 2-53
comparing files 2-36
compatibility of DOS mode with DOS 3.3 B-1
COMSPEC environment variable 2-35
COM0x.SYS device driver 4-17
CONFIG.SYS file 4-1
CONFIG.SYS statements 4-4
BREAK 4-5
BUFFERS 4-7
CODEPAGE 4-8
COUNTRY 4-10
DEVICE 4-13
DEVINFO 4-30
DISKCACHE 4-34
FCBS 4-36
IOPL 4-38
LIBPATH 4-39
MAXWAIT 4-41
MEMMAN 4-42
PAUSEONERROR 4-44
PRIORITY 4-45
PROTECTONLY 4-46
PROTSHELL 4-47
REM (remarks) 4-49
RMSIZE 4-50
RUN 4-51
SHELL 4-53
SWAPPATH 4-55
THREADS 4-56
TIMESLICE 4-57
TRACE 4-59
configuring the system 4-1
control keys 2-2
COPY command 2-39, 3-2
copying
all files on a diskette 2-56
archived files 2-152
changed files 2-151
diskettes 2-56
entire tree structure 2-151
files 2-39

copying *(continued)*
groups of files 2-151
subdirectories 2-151
system files 2-70
correcting disk errors 2-28
correcting file errors 2-28
countries
code pages 4-11
country codes 4-11
keyboards for 2-82
country code
Australia 4-11
Belgium 4-11
Canada (English) 4-11
Canada (French) 4-11
Denmark 4-11
Finland 4-11
France 4-11
Germany 4-11
Italy 4-11
Latin America 4-11
Netherlands 4-11
Norway 4-11
Portugal 4-11
Spain 4-11
Sweden 4-11
Switzerland 4-11
United Kingdom 4-11
United States 4-11
COUNTRY statement in CONFIG.SYS 4-10
CREATEDD command 2-43
creating a .BAT file 3-8
creating a .CMD file 3-8
creating a batch file 3-2
creating a directory 2-86
creating a dump diskette 2-43
creating a file with EDLIN C-16
Ctrl+Break 2-20, 4-5
Ctrl+Break keys 2-20
Ctrl+Num Lock keys 2-2
Ctrl+Z keys 2-39, C-25
current directory
changing 2-26
displaying the name of 2-26
customer service diskette 2-43

D

DATE command 2-45
date format 4-10
decimal separator symbol 4-10
defining code pages 4-8
DEL command (see ERASE)
deleting a directory 2-115
deleting a joined drive 2-79
deleting a substituted drive 2-133
deleting files 2-61
deleting the disk name 2-84
Denmark
code pages supported 4-11
country code 4-11
keyboard 2-81
DETACH command 2-47
detaching interactive processes 5-9
detaching programs 2-47
device drivers 4-13
ANSI.SYS 4-16
COM0x.SYS 4-17
DOS mode 1-4
EGA.SYS 4-19
EXTDSKDD.SYS 4-20
MOUSExxx.SYS 4-24
OS/2 mode 1-4
POINTDD.SYS 4-27
specifying path and filename 4-13
VDISK.SYS 4-28
device drivers in OS/2 and DOS modes 4-13
device error messages G-1
device errors, recovering from G-2
device installation warning messages 4-14
device names, reserved 2-2
device operation modes 2-88
device redirection 5-2
piping 5-2

DEVICE statement in
CONFIG.SYS 4-13
devices, preparing for code page
switching 4-30
DEVINFO statement in
CONFIG.SYS 4-30
differences
DOS mode and IBM Personal
Computer DOS 3.3 B-1
OS/2 mode and DOS mode com-
mands A-1
DIR command 2-49
directing input and output 5-2
piping 5-2
directories
changing 2-26
created by system
installation H-1
displaying files in 2-49, 2-143
displaying name of 2-26
searching for data files 2-8,
2-59
searching for programs and com-
mands 2-101
directory append 2-8
APPEND 2-8
disk 2-17
backing up files 2-17
checking for errors 2-28
displaying name of 2-149
formatting 2-70
naming 2-71, 2-84
organizing 2-86
partitioning a fixed disk 2-64
restoring backed up files 2-112
restoring files 2-112
disk cache support 4-34
disk status report 2-29
DISKCACHE statement in
CONFIG.SYS 4-34
DISKCOMP command 2-53
DISKCOPY command 2-56
diskette and drive
compatibility 2-54, 2-57
diskette comparing 2-53
diskette copying 2-56
diskettes for Operating
System/2 H-1
display adapters in DOS
mode 2-74, B-8
display adapters supporting code
page switching 4-31
display mode, setting 2-93
displaying
ASCII characters 128 through
125 2-74
code pages 2-22
directories 2-143
disk name 2-149
environment values 2-117
file contents 2-144
filenames in directory 2-49
joined drives 2-79
lines C-19
name of current directory 2-26
paths to data files 2-8, 2-59
paths to executable files 2-101
remarks 3-26
subdirectories 2-143
system prompt settings 2-105
version number of OS/2 2-146
volume label 2-149
DOS mode 2-74
command processor 1-3
default command prompt 2-106
device drivers 1-4
displaying extended character
set 2-74
editing with EDLIN C-1
environment variables 2-119
extended display and keyboard
support 4-16
printing B-6
starting for the first time 3-5
storage for programs 1-3
symbol for command 1-1
DOS mode only commands
APPEND 2-8

DOS mode only commands *(continued)*
ASSIGN 2-12
BREAK 2-20
COMMAND 2-34
GRAFTABL 2-74
JOIN 2-79
SETCOM40 2-121
SUBST 2-133
DOS programs B-3
DOS 3.3 and DOS mode differences B-1
DPATH command 2-59
dummy parameters 3-8
dump diskette 2-43, 2-44
dynamic link libraries 4-39

E

ECHO batch command 3-13
echoing 3-10, 5-10
of batch file statements 3-10
of redirection statements 5-10
editing with EDLIN C-1
backup file C-1
binary file C-3
existing file C-1
new file C-2
EDLIN C-1
EDLIN commands
A (append lines) C-8
C (copy lines) C-9
D (delete lines) C-10
E (end edit) C-15
edit line C-13
I (insert lines) C-16
L (list lines) C-19
M (move lines) C-22
P (page) C-23
Q (quit edit) C-24
R (replace text) C-25
S (search text) C-28
T (transfer lines) C-31
W (write lines) C-32
EDLIN system editor
command parameters C-4
commands C-6
common information C-6
prompt C-2
starting C-1
EGA.SYS device driver 4-19
end-of-file mark 2-39
ending a storage dump 2-44
ENDLOCAL batch command 3-15
environment string 2-117
environment values 2-117
environment variables 2-117, 3-6
ERASE command 2-61
erasing a directory 2-115
erasing files 2-61
error message, device G-1
error messages from CHKDSK 2-28
error messages without online help G-1
error messages, online help for 2-76
error recovery with dump diskette 2-44
EXIT command 2-63
exiting a program 2-20
EXTDSKDD.SYS device driver 4-20
extended display and keyboard support
DOS mode 4-16
OS/2 mode 2-7
extended partitions 2-64
extensions for EDLIN files
.BAK C-2, C-15, C-24
$$$ C-1
external commands
ANSI 2-7
APPEND 2-8
ASSIGN 2-12
ATTRIB 2-14
BACKUP 2-17
CHKDSK 2-28
CMD 2-32
COMMAND 2-34

external commands *(continued)*
COMP 2-36
CREATEDD 2-43
definition 2-1
DISKCOMP 2-53
DISKCOPY 2-56
FDISK 2-64
FIND 2-67
FORMAT 2-70
GRAFTABL 2-74
HELP 2-76
JOIN 2-79
KEYB 2-81
LABEL 2-84
MODE 2-88
MORE 2-97
PATCH 2-98
PRINT 2-103
RECOVER 2-107
REPLACE 2-110
RESTORE 2-112
SETCOM40 2-121
SORT 2-124
SPOOL 2-126
SUBST 2-133
SYS 2-135
TRACE 2-139
TRACEFMT 2-141
TREE 2-143
XCOPY 2-151
EXTPROC batch command 3-17

F

FCB (see file control block)
FCBS statement in
CONFIG.SYS 4-36
FDISK command 2-64
file attributes, changing 2-14
file control block 4-36
file IDs 4-36, 5-5
filename displaying 2-49
filename extensions for EDLIN
.BAK C-2, C-15, C-24
filename extensions for EDLIN *(continued)*
$$$ C-1
filename in syntax diagram 2-4
filename listing 2-49
filenames, specifying with DEVICE statements 4-13
files
backing up 2-17
combining 2-39
comparing 2-36, 2-53
copying 2-39, 2-151
copying all on a diskette 2-56
displaying contents of 2-144
displaying names of 2-49, 2-143
editing with EDLIN C-1
erasing 2-61
listing in a directory 2-51
organizing 2-86
printing 2-103, 2-126
renaming 2-109
restoring backed up 2-112
sorting contents of 2-125
verifying data in 2-28, 2-147
filter commands
definition 5-3
names of 5-3
FIND command 2-67
finding dynamic link libraries 4-39
finding file differences 2-36
finding matching lines 2-67
finding strings 2-67
Finland
country code 4-11
keyboard 2-81
fixing disk errors 2-28
fixing file errors 2-28
FOR batch command 3-18
foreign languages
code pages for 4-11, F-1
country dependent information for 4-10
keyboard layouts for 2-81
special characters 2-82

FORMAT command 2-70
formatting a dump diskette 2-43
formatting diskettes 2-56
formatting fixed disks and diskettes 2-70
formatting to less than drive capacity 2-70
France
  code pages supported 4-11
  country code 4-11
  keyboard 2-81

G

Germany
  code pages supported 4-11
  country code 4-11
  keyboard 2-81
global filename characters 2-2
GOTO batch command 3-20
GRAFTABL command 2-74
grouping commands with parentheses 5-4

H

HELP command 2-76
help with warning and error messages 2-76
hidden files 2-115
high capacity diskette formatting 2-70

I

IF batch command 3-21
input and output
  piping 5-2
  redirecting 5-1, 5-5
  with file control blocks in DOS mode 4-36
inserting lines with EDLIN C-1
installation of devices 4-14
installation program
  AUTOEXEC.BAT file 3-5
  default CONFIG.SYS settings 4-2
  directory structure created by H-1
  OS2INIT.CMD default file 3-4
internal commands
  BREAK 2-20
  CHCP 2-22
  CHDIR 2-26
  CLS 2-31
  COPY 2-39
  DATE 2-45
  definition 2-1
  DETACH 2-47
  DIR 2-49
  DPATH 2-59
  ERASE 2-61
  EXIT 2-63
  MKDIR 2-86
  PATH 2-101
  PROMPT 2-105
  RENAME 2-109
  RMDIR 2-115
  SET 2-117
  START 2-130
  TIME 2-137
  TYPE 2-144
  VER 2-146
  VERIFY 2-147
  VOL 2-149
internal file IDs 5-5
interrupting batch file processing 3-3
IOPL statement in CONFIG.SYS 4-38
Italy
  code pages supported 4-11
  country code 4-11
  keyboard 2-81

J

JOIN command 2-79
joining drives 2-79

K

KEYB command 2-81
keyboard layouts
  default layout 2-82
  for countries 2-82
keyboard templates D-1
  Belgium AT D-2
  Belgium Enhanced PC D-2
  Canada AT D-3
  Canada Enhanced PC D-3
  Denmark AT D-4
  Denmark Enhanced PC D-4
  France AT D-5
  France Enhanced PC D-5
  Germany AT D-6
  Germany Enhanced PC D-6
  Italy AT D-7
  Italy Enhanced PC D-7
  Latin America AT D-8
  Latin America Enhanced PC D-8
  Netherlands Enhanced PC D-9
  Norway AT D-10
  Norway Enhanced PC D-10
  Portugal AT D-11
  Portugal Enhanced PC D-11
  Spain AT D-12
  Spain Enhanced PC D-12
  Sweden/Finland AT D-13
  Sweden/Finland Enhanced PC D-13
  Switzerland AT D-14
  Switzerland Enhanced PC D-14
  United Kingdom AT D-15
  United Kingdom Enhanced PC D-15
  United States AT D-16
  United States Enhanced PC D-16
keys
  accent 2-82
  accent characters E-1
  for foreign languages 2-82, E-1
keyword command parameters 2-2

L

LABEL command 2-84
language code pages
  limitation of 2-23
  multilingual 2-23
languages, foreign
  code pages for 4-11, F-1
  country dependent information for 4-10
  keyboard layouts for 2-81
Latin America
  code pages supported 4-11
  country code 4-11
  keyboard 2-81
leap year 2-45
LIBPATH statement in CONFIG.SYS 4-39
line editor C-1
listing filenames 2-49
listing files in a directory 2-51
listing lines with EDLIN C-23
listing selected files 2-51
locating data files 2-8, 2-59
locating dynamic link libraries 4-39
locating matching lines 2-67
locating programs and commands 2-101
locating strings 2-67
log file (backup) 2-17

M

making a directory 2-86
MAXWAIT statement in
CONFIG.SYS 4-41
MD 2-86
MEMMAN statement in
CONFIG.SYS 4-42
message id 2-78
message identifier 2-77
messages without online help G-1
messages, online help for 2-76
MKDIR command 2-86
MODE command 2-88
mode, DOS 1-1
mode, OS/2 1-1
modems B-3
modems, DOS mode B-8
MORE command 2-97
mouse device driver 4-24
mouse device drivers, DOS
mode B-8
mouse installation 4-24
MOUSExxx.SYS device driver 4-24
multilingual code page 2-23, F-3
switching to with CHCP
command 2-23
multitasking 1-1

N

naming batch files 3-1
naming disk during format 2-71
national language code
pages 2-23, 4-11
nesting batch files 3-3, 3-11
Netherlands
code pages supported 4-11
country code 4-11
keyboard 2-81
nonescaping key combinations E-1
nonescaping keys 2-82
noninteractive processing 2-47
noninteractive programs 5-9
Nordic code page F-6
Norway
code pages supported 4-11
country code 4-11
keyboard 2-81

O

operating modes
command differences 1-1, A-1
DOS 1-1
OS/2 1-1
operating several programs 1-2
optional command parameters 2-3
OR command operator (||) 5-4
OR command operator (¦¦) 5-4
OS/2 diskettes H-1
OS/2 mode
command processor 1-2
default command prompt 2-106
definition 1-3
device drivers 1-4
environment variables 2-119
extended display and keyboard
support 2-7
multitasking 1-2
OS2INIT.CMD 3-4
starting CMD.EXE
automatically 3-5
STARTUP.CMD 3-5
storage for programs 1-3
OS/2 mode only commands
ANSI 2-7
CMD 2-32
CREATEDD 2-43
DETACH 2-47
DPATH 2-59
FDISK 2-64
KEYB 2-81
SPOOL 2-126
START 2-130
TRACE 2-139
TRACEFMT 2-141
OS/2 system, transferring 2-135
OS/2 version number 2-146

output
piping 5-2
preventing 5-9
redirecting 5-1
with file control blocks in DOS mode 4-36
overview
batch commands A-6
CONFIG.SYS statements A-7
operating modes 1-1
OS/2 commands A-1
running DOS programs 1-3
running OS/2 programs 1-2
storage allocation 1-2

P

parallel printer mode 2-94
parameters
EDLIN C-4
replaceable 3-8
partitioning the fixed disk 2-64
partitions
active 2-64
changing 2-64
creating 2-64
deleting 2-64
displaying 2-64
PATCH command 2-98
patching in automatic mode 2-99
PATH command 2-101
paths to data files in DOS mode 2-8
paths to data files in OS/2 mode 2-59
paths to device drivers 4-13
paths to executable files 2-101
PAUSE batch command 3-24
PAUSEONERROR statement in CONFIG.SYS 4-44
percent sign (%) in batch files 3-8
piping
definition 5-2
symbol for (|) 5-2
piping *(continued)*
symbol for (¦) 5-2
plotters, DOS mode B-9
plotting in DOS mode B-6
POINTDD.SYS 4-27
pointing device 4-24
Portugal
code pages supported 4-11
country code 4-11
keyboard 2-81
Portuguese code page F-4
preparing devices for code page switching 4-30
preventing output 5-9
primary partitions 2-64
PRINT command 2-103
print queue 2-103
print spooler 2-103, 2-126
printers supporting code page switching 4-31
printing 2-103
printing characters 2-94
printing in DOS mode 2-129, B-6
printing 6 or 8 lines per inch 2-94
priority calculation 4-45
absolute 4-45
dynamic 4-45
PRIORITY statement in CONFIG.SYS 4-45
problem determination
with CHKDSK command 2-28
with service programs 2-6
processing noninteractive programs 5-9
program selector 4-47
programs written for IBM Personal Computer DOS B-3
PROMPT command 2-105
Proprinter 4-33
PROTECTONLY statement in CONFIG.SYS 4-46
PROTSHELL statement in CONFIG.SYS 4-47

punctuation in commands 2-2

Q

querying code pages 2-22
quick reference to
  batch commands A-6
  OS/2 commands A-1
  statements for CONFIG.SYS A-7
Quietwriter III printer 2-129, 4-33
quitting a program,BREAK
  command 2-20

R

RD 2-115
read-only attribute of a file 2-14
reading command diagrams 2-2
reconfiguring the system 4-2
RECOVER command 2-107
recovering files 2-107
recovering from device errors G-2
redirecting input and output 5-2
redirection statements,
  echoing 5-10
redirection symbols
  < (redirect input) 5-2
  > (redirect output) 5-1
  >> (append redirected
    output) 5-2
reinstalling the system 4-2
REM (remarks)
  batch command 3-26
  statement in CONFIG.SYS 4-49
remarks
  in batch file 3-26
  in CONFIG.SYS file 4-49
removing a directory 2-115
removing files 2-61
REN 2-109
RENAME command 2-109
renaming files 2-109
REPLACE command 2-110
replaceable parameters 2-117,
  3-8, 3-9
reserved device names 2-2
responses to the system G-2
RESTORE command 2-112
restoring backed up files 2-112
restoring subdirectories 2-113
retry read/write operation G-2
reverse sort 2-124
RMDIR command 2-115
RMSIZE statement in
  CONFIG.SYS 4-50
rules for commands 2-1
RUN statement in
  CONFIG.SYS 4-51
running batch files 3-2
running noninteractive
  programs 5-9

S

screen interface 4-47
searching directories
  for data files 2-8, 2-59
  for programs and
    commands 2-101
searching for matching lines 2-67
searching for strings 2-67
segment combining 1-3
segment swapping 1-3, 4-42
separating commands with the &
  symbol 5-4
serial device driver 4-17
serial devices
  device driver for 4-17
  setting operation mode for 2-89
  setting the DOS port
    address 2-121
  using in DOS mode B-6
serial devices in DOS mode 2-121,
  B-4, B-6
service programs 2-6
SET command 2-117
SETCOM40 command 2-121
SETLOCAL batch command 3-27
setting a search path
  for data files 2-8, 2-59

setting a search path *(continued)*
for programs and commands 2-101
setting a search path to null
APPEND 2-8
DPATH 2-59
PATH 2-101
setting asynchronous communications mode 2-89
setting characters per line 2-94
setting display mode 2-93
setting environment 2-117
setting lines per inch 2-94
setting parallel printer mode 2-94
setting vertical spacing 2-94
SHELL statement in CONFIG.SYS 4-53
SHIFT batch command 3-28
showing a search path
APPEND 2-8
DPATH 2-59
PATH 2-101
showing ASCII characters 128 thru 255 2-74
showing environment 2-117
showing joined drives 2-79
showing name of current directory 2-26
showing the prompt 2-105
SORT command 2-124
sorting a file 2-124
Spain
code pages supported 4-11
country code 4-11
keyboard 2-81
special batch files 3-4
SPOOL command 2-126
spooling printer files 2-126
stand-alone dump 2-44
standard input 5-5
standard output 5-5
START command 2-130, 3-6
using in STARTUP.CMD 3-6
starting a storage dump 2-44
starting OS/2 programs automatically 3-6
starting programs automatically
AUTOEXEC.BAT 3-5
OS2INIT.CMD 3-4
STARTUP.CMD 3-5
starting the system 1-4
STARTUP.CMD file 3-5
statements in CONFIG.SYS 4-4
stopping a program 2-20
storage allocation 1-2
storage compaction 1-3, 4-42
storage dump 2-44
storage overcommitment 1-3
storage restrictions for EDLIN files C-2
subdirectories
backing up 2-18
copying 2-151
restoring 2-113
SUBST command 2-133
substituting a drive letter for a drive and path 2-133
suppressing echoing 3-10
SWAPPATH statement in CONFIG.SYS 4-55
swapping segments 1-3
Sweden
code pages supported 4-11
country code 4-11
keyboard 2-81
switching code pages 2-22, 2-23
interrelated statements in CONFIG.SYS 4-3
purpose of 2-23
Switzerland
code pages supported 4-11
country code 4-11
Swiss-French keyboard 2-81
Swiss-German keyboard 2-81
symbols
< (redirect input) 5-2
() (command grouper) 5-4

symbols *(continued)*
| (piping symbol) 5-2
|| (OR command operator) 5-4
& (command separator) 5-4
&& (AND command operator) 5-3
> (redirect output) 5-1
>> (append redirected output) 5-2
¦ (piping symbol) 5-2
¦¦ (OR command operator) 5-4
decimal separator 4-10
for DOS command 1-1
for OS/2 command 1-1
syntax in command diagrams 2-2
syntax of commands 2-2
SYS command 2-135
system code pages 4-8
defining in CONFIG.SYS 4-8
names of 4-8
system date 2-45
system editor C-1
system files, copying during format 2-70
system initialization 1-4
pausing for error display 4-44
system installation
system installation program
AUTOEXEC.BAT file 3-5
default CONFIG.SYS settings 4-2
directory structure created by H-1
OS2INIT.CMD default file 3-4
system limitations, DOS mode B-8
system startup 1-4
system time 2-137
system tracing 2-139, 4-59, 4-61
system transfer 2-135

T

table of
batch commands A-6
CONFIG.SYS statements A-7
OS/2 commands A-1
templates, Keyboard D-1
threads 4-56
definition 4-56
THREADS statement in CONFIG.SYS 4-56
time
displaying 2-137
setting 2-137
TIME command 2-137
time format 4-10
TIMESLICE statement in CONFIG.SYS 4-57
TRACE
OS/2 command 2-139
statement in CONFIG.SYS 4-59
TRACEBUF statement in CONFIG.SYS 4-61
TRACEFMT command 2-141
transferring system files 2-135
TREE command 2-143
TYPE command 2-144

U

U.S. IBM PC code page F-2
United Kingdom 2-82
code pages supported 4-11
country code 4-11
keyboard 2-81, 2-82
United States
code pages supported 4-11
country code 4-11
keyboard 2-81
updating the system 2-135
user interface 4-47

V

variable command parameters 2-2
VDISK.SYS 4-28
VER command 2-146
VERIFY command 2-147
verifying copy 2-152
verifying data in files 2-147
verifying during copy 2-40
version number 2-146
vertical spacing 2-94
viewing a file 2-144
viewing long files 2-97
VOL command 2-149
volume label (disk name)
  changing 2-84
  creating during format 2-71
  deleting 2-84
  viewing 2-149

W

warning messages during device installation 4-14
warning messages, online help for 2-76

X

XCOPY command 2-151